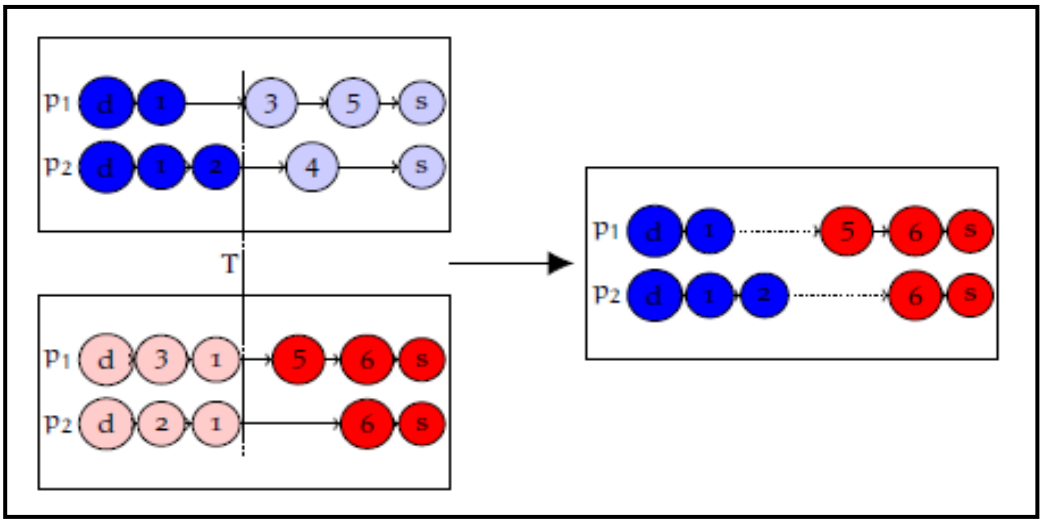
# Selective vehicle routing problems during wildfires

Quentin Peña

Par **Quentin PENÃ**

# *Selective vehicle routing problems during wildfires*

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC




Soutenue le 11 décembre 2023
**Spécialité :** Informatique : Unité de recherche Heudyasic (UMR-7253)
D2780




UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE

DOCTORAL THESIS

---

# Selective vehicle routing problems during wildfires

---

*Author:* **Quentin PENÃ**

Spécialité : Informatique

:

Laboratoire Heudiasyc

Thesis defended on December 11, 2023

*Supervisors:* Aziz Moukrim and Mehdi Serairi

*Reviewers:*

Hamid ALLAOUI
Professor
University of Artois

Caroline PRODHON
Professor
Univ. of Technology of Troyes

*Examiners:*

Taha ARBAOUI
Associate Professor
Univ. of Technology of Troyes

Dritan NACE
Professor
Univ. of Technology of Compiègne

*A thesis submitted in fulfillment of the requirements for the degree of Doctor*

*in the*

SCOP (Sûreté, Communication, Optimisation) Team
Heudiasyc Laboratory

# *Remerciements*

Je voudrais tout d'abord exprimer ma plus sincère gratitude à mes co-directeurs de thèse, MM. Aziz Moukrim et Mehdi Serairi, qui m'ont fait l'honneur de superviser ma thèse. Leur expertise, leurs conseils éclairés et leur dévouement ont été essentiels à la réalisation de cette thèse. Je suis reconnaissant pour l'opportunité qui m'a été donnée de travailler sous leur direction, et je ne saurais les remercier suffisamment pour leur patience et leur encouragement constant.

Je voudrais ensuite remercier M. Hamid Allaoui et Mme Caroline Prodhon d'avoir accepté d'être les rapporteurs de ce manuscrit. J'étends ces remerciements à MM. Taha Arbaoui et Dritan Nace pour avoir accepté de participer au jury de soutenance.

Je tiens à exprimer ma sincère reconnaissance envers les membres du laboratoire Heudiasyc, en particulier les membres de l'équipe SCOP, pour leur accueil chaleureux. Les conditions dans lesquelles la thèse s'est déroulée n'ont pas permis de profiter pleinement de la vie au laboratoire pendant de longs mois, mais les rencontres et les discussions ont été enrichissantes, tant sur le plan personnel que scientifique et professionnel. J'ai une pensée particulière pour mes collègues de bureau, Mohamed Amine Ouberkouk et Lahcene Mezouari, désormais docteurs.

Je voudrais également remercier l'Ecole Doctorale pour leur soutien académique, ainsi que l'ensemble de l'administration de l'Université de Technologie de Compiègne.

Pour terminer, je tiens également à remercier ma famille et mes amis d'avoir été là, pour leur soutien constant, pendant la thèse, mais déjà bien avant.

UNIVERSITY OF TECHNOLOGY OF COMPIÈGNE

# *Abstract*

SCOP (Sûreté, Communication, Optimisation) Team
Heudiasyc Laboratory

Doctor

**Selective vehicle routing problems during wildfires**

by Quentin Peña

In the last 10 years, around 82 million hectares of forests have been destroyed by wildfires. These fires pose imminent threats to the wildlife, as well as in urban areas, to human lives and critical infrastructure. The need for a strategic and resource-efficient response to the growing threat of wildfires, exacerbated by climate change, led to the launch of the GEO-SAFE project. Among its many challenges, this thesis focuses on the Asset Protection Problem during escaped wildfires (APP) and its disrupted counterpart, D-APP. These problems aim to route firefighting crews to carry out protection operations on community assets. Existing exact methods, particularly the Mixed Integer Program (MIP) for APP, struggled with efficiency. We performed a structural analysis that led to significant enhancements, notably the introduction of three sets of valid inequalities. These improvements not only decreased solution times but also stabilized an exact solution method for smaller instances. In parallel, D-APP's complex replanning requirements for larger instances were met through a novel reformulation based on a dominance relation on the solutions of the problem. Acknowledging the limitations of exact methods in real-time scenarios, this thesis introduces two heuristic approaches tailored for D-APP. The first, based on the new formulation, uses a relax-and-fix approach, while the second implements a genetic algorithm. These methods generated good approximate solutions in a limited time, paving the way for rapid and effective wildfire response strategies. Through these contributions, this work stands as a testament to the ongoing battle against wildfires, reinforcing the synergy between computational techniques and real-world disaster management.

**Keywords:** Wildfire Response; Asset Protection; Mixed Integer Programming; Heuristic Algorithms; Bi-objective Optimization; Disruption Management

UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

# *Résumé*

Equipe SCOP (Sûreté, Communication, Optimisation)
Laboratoire Heudiasyc

Doctorat

**Problèmes d'acheminement sélectif des véhicules lors de feux de forêt**

par Quentin Peña

Lors de la dernière décennie, environ 82 millions d'hectares de végétation ont été détruites dans des feux de forêt. Ces incendies mettent en danger la nature mais aussi, dans les zones urbaines, des vies humaines et les infrastructures. Le danger croissant des incendies, exacerbé par le changement climatique, incite au développement d'outils stratégiques pour y répondre efficacement, notamment par le biais du projet GEO-SAFE. Parmi les nombreuses problématiques liées, les travaux de cette thèse se concentrent sur le problème de protection des biens (APP), et sa version perturbée, D-APP. Ces problèmes visent à déployer des flottes de pompiers pour assurer des opérations préventives de protection. Les méthodes de résolution exactes existantes, en particulier de programmation linéaire en nombre entiers (PLNE) pour APP, ne sont pas efficaces. Nous avons étudié la structure du problème afin d'obtenir de franches améliorations, notamment via l'introduction de trois ensembles d'inégalités valides. Ces améliorations ont non seulement grandement réduit les temps de résolution, ils ont permis d'avoir une méthode de résolution stable pour les petites instances. En parallèle, une nouvelle formulation PLNE de D-APP, plus efficace, a été proposée, basée sur une relation de dominance entre les solutions. Toutefois, les méthodes de résolution exactes sont mal adaptées à des scénarios en temps réel : cette thèse introduit également deux méthodes de résolution heuristique adaptées à notre problème. La première, reposant sur la reformulation, utilise une approche *relax-and-fix*. La seconde implémente un algorithme génétique. Ces méthodes ont généré des solutions approchées de bonne qualité en temps limité, permettant de déployer des réponses rapidement et efficacement. Á travers ces contributions, ces travaux s'inscrivent dans la lutte contre les incendies, en renforçant la synergie entre techniques d'optimisation et gestion de crises.

**Mots-clés:** incendies ; protection des biens ; programmation linéaire ; résolution heuristique ; optimisation bi-objectif ; perturbation

*To my great-grandmother, Marie*

# Contents

# List of Tables

# List of Figures

Chapter 1

# Introduction

## Contents



### 1.1 Background

In recent years, the world has witnessed a surge in devastating wildfires, a phenomenon exacerbated by the effects of climate change. Beyond the wilderness, these wildfires have jeopardized urban areas, posing imminent threats to both lives and critical infrastructure. The escalation of such incidents necessitates a strategic and resource-efficient response, where the deployment of firefighting teams is pivotal. This response phase not only involves active suppression efforts but also the evacuation of people and safeguarding valuable goods.

Recognizing the urgency of this situation, the Geospatial Based Environment For Optimisation Systems Addressing Fire Emergencies (GEO-SAFE) project was initiated in 2016. This collaborative project brought together European and Australian research teams, uniting their expertise to develop innovative tools facilitating decision-making during the response phase. Spanning multiple disciplines, the GEO-SAFE project delved into various challenges, ranging from risk prevention to life and goods protection, as well as advancements in cartography.

Within the scope of this manuscript, our primary focus is the protection of goods during a wildfire. Incident Management Teams are faced with complex challenges, aiming to deploy resources efficiently to protect community assets. Central to this effort is the Asset Protection Problem during escaped wildfires (APP), a challenge first introduced by Merwe et al. (2015). APP is a selective vehicle routing problem containing intricate constraints such as time windows, synchronization, and cooperation requirements.

Yet, emergency situations are inherently unpredictable, introducing disruptions that can render initial plans obsolete. For APP, such disruptions include vehicle breakdowns or road closures. In the context of APP, these disruptions can be vehicle breakdowns or road closures, demanding rapid adaptation and reconfiguration of response strategies.

This gives rise to the Disrupted Asset Protection Problem (D-APP) as outlined by Merwe et al. (2017).

In this thesis, we address both versions of the problem. Notably, the existing Mixed Integer Program (MIP) proposed for APP struggles to generate optimal solutions for modest instances within the literature. To enhance the efficiency of this formulation, we lead a structural analysis of APP, identifying valid inequalities that substantially improve MIP performance and decrease solution times. Simultaneously, we conduct a similar investigation for D-APP, adapting the valid inequalities and proposing a more effective formulation. The valid inequalities and reformulation [Peña et al., 2023] have been submitted for publication. While obtaining optimal solutions for D-APP is crucial for evaluating deployed strategies, real-time emergency scenarios require solutions in minutes, making exact methods impractical. To address this challenge, we introduce two heuristic approaches: a relax-and-fix algorithm, and an implementation of a genetic algorithm (NSGA-II). These methods offer rapid, approximate solutions, ensuring a timely and effective response in the face of unpredictable disruptions. The results obtained from the genetic algorithm have been presented at the 15th International Conference on Artificial Evolution [Peña et al., 2022]. Through these efforts, this thesis aims to enhance our understanding of wildfire response strategies, contributing to broader disaster management and community safety initiatives.

This study was carried out within the framework of GEOSAFE (Geospatial Based Environment For Optimization Systems Addressing Fire Emergencies). This work was partially supported by the framework of the Labex MS2T, funded by the French Government, via the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

## 1.2 THESIS OUTLINE

This thesis is organized as follows. In Chapter 2, the first part provides an overview of the challenges related to responding to wildfires. We review multiple issues highlighted by various works within the GEO-SAFE project. In the second part, we introduce the different optimization problems related to vehicle routing and disruption management. In the third part, we provide basic definitions related to multi-objective optimization, with a particular focus on bi-objective optimization. We also recall the most popular methods presented in the literature to solve them.

In Chapter 3, we tackle the Asset Protection Problem during escaped wildfires (APP), that aim at maximizing the total protected value of deployed firefighting vehicles, subject to time windows, cooperation and synchronisation constraints. We study the structure of the problem and introduce three sets of valid inequalities, based on resources and time windows, to improve its MIP formulation. The valid inequalities manage to stably decrease the solution time under a minute for small literature instances, down from potentially hours.

In Chapter 4, we focus on the disrupted version of APP, D-APP. This bi-objective problem aims at generating efficient routes after a disruption invalidates the initial routes. We propose a new MIP formulation, based on a dominance relation on the solutions of

the problem, as well as an extension of APP's valid inequalities. The reformulation clearly outperforms the initial formulation and, associated with the valid inequalities, manages to solve larger instances in reasonable time.

In Chapter 5, we shift our focus to heuristic approaches to solve D-APP. We propose two heuristic methods tailored to our problem. The first uses the new formulation for D-APP in a relax-and-fix algorithm. The second implements a widely adopted genetic algorithm for multi-objective optimization. Each method has its advantages and offer high-quality approximate fronts in constrained time.

Finally, in Chapter 6, we provide a conclusion that summarizes our contributions and outlines some perspectives for our future work.

# Chapter 2

# General context

## Contents



### 2.1 Introduction

During escaped wildfires, urban areas are at risk of destruction, which can be detrimental to the communities and infrastructure. This manuscript takes particular interest in swift and efficient deployments of emergency response teams in order to protect community assets, and adapting plans to unforeseen disruptions.

This chapter introduces the general context of the problems we consider in this manuscript. Firstly, in Section 2.2, we provide a summary of some of the main challenges of decision-making during fire emergencies, with a particular focus on the work achieved through the Euro-Australian project GEO-SAFE. Then, in Section 2.3, we present an overview of selective vehicle routing problems. Lastly, in Section 2.4, we introduce the basic concepts of bi-objective optimization as well as exact solution methods to solve them. A general conclusion of the chapter is provided in Section 2.5.

## 2.2 FIRE EMERGENCIES

In recent years, wildfires have been more frequent and more difficult to control. Due to global warming, summers are drier and hotter, favoring the propagation and strength of such fires [Running, 2006]. Millions of hectares of vegetation are burnt each year, and lives and properties are damaged or lost all over the world, from Europe [Pausas et al., 2009] to the United States [Short, 2014] and including Australia [Cruz et al., 2012].

According to the Federal Emergency Management Agency (FEMA), the emergency management process consists of four distinct phases: Mitigation, Preparedness, Response, and Recovery.

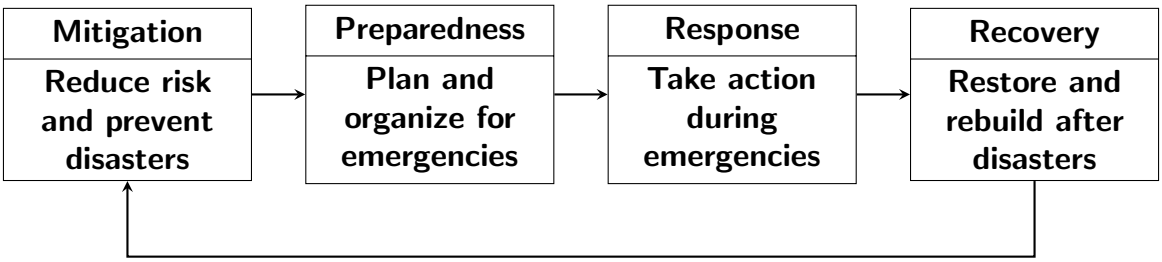



Figure 2.1: Emergency management phases

The Mitigation phase involves taking proactive measures to prevent or minimize the causes, impact, and consequences of disasters. Some examples of hazard mitigation include securing homes or barns with ground anchors to withstand wind damage, creating water channels and planting vegetation to redirect and absorb water, constructing levees or barriers to control flooding, reinforcing fencing to prevent animal escapes, and obtaining insurance policies to mitigate financial risks.

The Preparedness phase focuses on planning, training, and educational activities for events that cannot be fully mitigated. This involves developing disaster preparedness plans that outline appropriate actions to take, where to go, and whom to contact in case of a disaster. It also includes conducting drills, tabletop exercises, and full-scale exercises to test and refine the plans. Creating a supply list of essential items for emergencies and identifying vulnerabilities to specific hazards, such as high winds on a farm, are also part of the preparedness phase.

The Response phase occurs immediately after a disaster and involves implementing the previously developed disaster response plans. During this phase, normal business and operational activities are disrupted. The effectiveness and duration of the response phase depend on the level of preparedness. Response activities include implementing response plans, conducting search and rescue missions, taking actions to ensure personal safety and protect others, and addressing public concerns regarding food safety in the aftermath of a disaster.

The Recovery phase encompasses the restoration efforts that take place concurrently with regular operations and activities. The recovery period following a disaster can be lengthy. Recovery activities include preventing or reducing stress-related illnesses and excessive financial burdens, rebuilding damaged structures based on lessons learned

from previous disasters, and reducing vulnerability to future disasters by implementing measures informed by advanced knowledge gained from past events.

The increasing frequency and intensity of wildfires demand effective response operations to safeguard ecosystems, human lives, and property. Faced with the many challenges rising from the increase in number and intensity of wildfires, scientific communities from Europe and Australia launched a common project: Geospatial Based Environment For Optimisation Systems Addressing Fire Emergencies (GEO-SAFE). The GEO-SAFE project emerges as a crucial initiative to advance fire management through innovative tools and collaborative knowledge exchange. In this section, we provide an overview of response operations to wildfires, with a specific focus on integrating insights from the GEO-SAFE project. By incorporating dynamic risk cartography, resource allocation tools, and training processes, we highlight the potential of the GEO-SAFE project to enhance the efficacy of wildfire response operations. The project spans multiple disciplines, from cartography to crisis management or operations research, for example. Fire management problems were split in three main categories:

- Fire suppression and fire propagation control.
- Life and goods protection.
- Implementation and training.

We propose in the following sections a deep dive into each category, highlighting the main challenges and contributions.

### 2.2.1 *Fire suppression and fire propagation control*

Active suppression measures play a pivotal role in controlling and mitigating the spread of wildfires. The GEO-SAFE project contributes to active suppression efforts by developing a dynamic risk cartography tool. This tool utilizes data collected from satellite and remote sensors to analyze and forecast fire extension, allowing for improved predictions of fire behavior and risk evolution during the response phase. By integrating the dynamic risk cartography into active suppression strategies, response teams can enhance their decision-making processes and resource allocation, leading to more effective fire suppression efforts.

The GEO-SAFE project also focuses on developing efficient algorithms for risk prevention. For example, León et al. (2019) and Croce et al. (2020) explore different aspects of fuel management. The landscape is divided into cells to be treated in order to mitigate the potential spread of wildfires. The objective is to select cells for treatment to minimize the presence of contiguous old cells throughout the time horizon.

### 2.2.2 *Life and goods protection*

Timely and safe evacuation of affected populations is critical in minimizing the risks posed by wildfires. Evacuation is a primary method of protecting lives, and it can be performed through self-evacuation, assisted evacuation, or supported evacuation.

Self-evacuation and assisted evacuation are often studied using simulation models or network flow models ([Artigues et al., 2019]). Supported evacuation, where people are evacuated by buses or collective transport, can be approached using optimization models to make decisions on necessary resources and shelter organization ([Flores et al., 2023]). The choice between evacuation and sheltering depends on factors such as perception, observation of others' actions, environmental cues, and wildfire information. Self-evacuation and assisted evacuation have been extensively studied (see [Ozdamar et al., 2017], [Veeraswamy et al., 2018]), while supported evacuation, where individuals require specific support, is less explored.

Preventive protection measures are essential in reducing vulnerabilities to wildfires. Incident Management Teams (IMT) play a crucial role in minimizing risks and protecting vital infrastructure during wildfires. However, IMTs face complex challenges and decision-making difficulties in their operations. In this context, the focus is on out-of-control wildfires spreading across a landscape and posing a threat to various assets such as bridges, electric substations, schools, and factories. Preventive activities performed by IMTs near these assets before the fire impact are essential to reduce the risk of damage. Defensive tasks include fuel material removal, wetting down buildings, and extinguishing fires. To address the problem of choosing which assets to protect and when, an orienteering problem formulation is proposed ([Merwe et al., 2015], [Merwe et al., 2017]).

### 2.2.3 *Implementation and training*

The project goes beyond the development of innovative tools by placing a strong emphasis on facilitating their implementation and providing comprehensive training support. By working closely with end-users and incorporating their feedback, the GEO-SAFE project ensures that the developed solutions align with their needs and operational realities. Furthermore, the project develops training tools to enhance the capabilities and preparedness of response teams. These training tools provide valuable guidance on the utilization of the dynamic risk cartography, resource allocation tools, and other innovations within the context of wildfire response operations. By emphasizing implementation and training, the GEO-SAFE project enables the practical application of developed solutions and fosters a shared culture of research and innovation, ultimately improving the overall effectiveness of response operations to wildfires.

## 2.3 RELATED PROBLEMS

This section describes the scientific context of this thesis. We provide an overview of vehicle routing problems as well as disruption management problems.

### 2.3.1 *Vehicle Routing Problems*

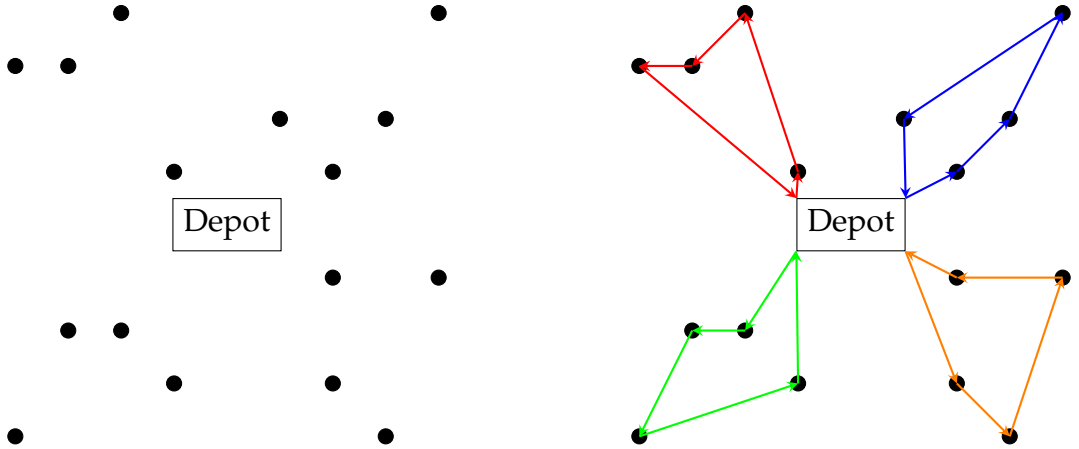
The Vehicle Routing Problem (VRP) has been a subject of extensive research for more than fifty years, mainly due to its significant applications in logistics and its inherent complexity. This combinatorial optimization problem involves efficiently planning routes at a minimal cost to serve a group of geographically distributed customers while respecting various constraints related to resources and customer demands.

The Traveling Salesman Problem (TSP), introduced by Robinson (1949), is one of the most studied routing problem. The problem arose from the necessity of a commercial salesman to strategically plan visits to a specific set of cities before ultimately returning to their starting point, in order to minimize the distance traveled. For the TSP, as well as most subsequent routing problems, graphs are used for modeling and solving the problem. Let consider a complete oriented and weighted graph $G = (V, A)$, with $V = \{v_0, v_1, ..., v_n\}$ representing the cities that must be visited by the salesman. Each arc $(v_i, v_j) \in A$ represents the path connecting the two cities $v_i$ and $v_j$, and is associated with a travel time $C_{ij}$. An optimal solution of the TSP is a shortest Hamiltonian cycle on the graph G.

Vehicle Routing Problems (VRP) generalize the TSP by considering multiple vehicles. Initially introduced as the "Truck Dispatching Problem" [Dantzig et al., 1959], the VRP has evolved throughout the years. Inspired by real-life applications, various different objectives and operational constraints have been introduced to VRP. Laporte (2009) introduced the classical VRP formulation. Let consider a complete undirected graph $G = (V \cup \{0\}, E)$, with $V = \{1, 2, .., n\}$ representing customers and vertex $0$ the depot. Each edge of the set $E = \{(i, j) : i, j \in V \cup \{0\}$ interconnecting the vertices is associated with a cost $C_{ij}$. The cost distribution is expected to satisfy the triangle inequality. A demand $d_i$ is associated with each customer $i \in V$. A fleet $F$ of $m$ homogeneous vehicles with a maximum capacity Q is available to serve the customers. A solution of VRP is a set of $m$ routes, that start and end at the depot, where each customer is visited exactly once. The total demand of the customer served on a route cannot exceed the maximum capacity Q. An optimal solution of the VRP is a set of routes serving all customers such that the total traveled cost is minimized. Figure 2.2 illustrates an instance of VRP and a possible solution.

In the following, we will give an overview of the formulations have been proposed to model the VRP, either using an exponential number of constraints or variables.

FLOW FORMULATIONS The first flow formulation is based on flow of vehicles. Introduced by Laporte et al. (1983). Binary variables $x_{ij}$, $(i, j) \in E$, are set to 1 if a vehicle of the fleet uses the edge $(i, j)$, 0 otherwise. The set $\delta^+(S)$ (respectively $\delta^-(S)$), with $S \subseteq V$, is the set of edges with $i \in S$ and $j \in V \setminus S$ (respectively $i \in V \setminus S$ and $j \in S$).




**(a)** A VRP instance **(b)** A VRP solution

Figure 2.2: Example of a Vehicle Routing Problem

$$\text{Minimize} \sum_{(i,j)\in E} C_{ij}x_{ij} \tag{2.1}$$

$$\sum_{j=1}^{n} x_{0j} \leqslant m \tag{2.2}$$

$$\sum_{(i,j)\in\delta^+(\{i\})} x_{ij} = 1 \quad \forall i \in V \tag{2.3}$$

$$\sum_{(i,j)\in\delta^-(\{i\})} x_{ij} = 1 \quad \forall i \in V \tag{2.4}$$

$$\sum_{(i,j)\in\delta^-(S} x_{ij}) \geqslant \left\lceil \frac{\sum_{i\in S} q_i}{Q} \right\rceil \quad \forall S \subseteq V \tag{2.5}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E \tag{2.6}$$

Equation (2.1) defines the objective function, that minimizes the total travel time. Constraints (2.2-2.4) are the flow constraints. These constraints ensure that at most $m$ vehicles leave the depot, and that exactly one arc is used to enter and leave (respectively) each customer. Constraints (2.5) exclude solutions that use less vehicles than the minimal number necessary to meet the capacity requirement for any set of assets. These constraints also prevent subtours, ie., cycles that do not go through the depot, from appearing in a solution. Constraints (2.6) define the domain of the decision variables.

The second flow formulation is based on flow of resources. Introduced by Gavish (1984). It added new continuous variables $f_{ij}$, $(i,j) \in E$, which represent the quantity of resources carried along the edge $(i,j)$. Constraints (2.5) are replaced by the following constraints:

$$\sum_{(i,j)\in d^-(\{i\})} f_{ij} - \sum_{(i,j)\in d^-(\{i\})} f_{ij} = q_i \quad \forall i \in V \tag{2.7}$$

$$q_i x_{ij} \leqslant f_{ij} \leqslant (Q - Q_i)x_{ij} \quad \forall (i,j) \in E \tag{2.8}$$

Constraints 2.7 ensure that each customer is correctly served. Constraints 2.8 are bounds on the value of variables $f_ij$.

PARTITIONING FORMULATION Introduced by Balinski et al. (1964). Let $R$ be the set of feasible routes. A binary coefficient $a_{ir}$ is set to 1 if customer $i \in V$ is visited by route $r \in R$. The total cost of arcs used by route $r$ is denoted $c_r$. Binary variables $x_r$ are set to 1 if route $r$ is used in the solution, 0 otherwise.

$$\text{Minimize} \sum_{r\in R} c_r x_r \tag{2.9}$$

$$\sum_{r\in R} a_{ir} x_r = 1 \quad \forall i \in V \tag{2.10}$$

$$\sum_{r\in R} x_r \leqslant m \tag{2.11}$$

$$x_r \in \{0,1\} \quad \forall r \in R \tag{2.12}$$

Equation (2.9) defines the objective function, that minimizes the total cost of the selected routes. Constraints (2.10) ensure that exactly one selected route visits each customer. Constraint (2.11) limits the number of routes that can be selected. Constraints (2.12) define the domain of the decision variables.

When cost distribution $c$ satisfy the triangle inequality, constraints (2.10) can be replaced by:

$$\sum_{r\in R} a_{ir} x_r \geqslant 1 \quad \forall i \in V \tag{2.13}$$

The formulation consisting of (2.9, 2.13, 2.11, 2.12) is referred to as the set-covering formulation. Optimal solutions of the partitioning and set-covering formulations have the same cost $c$. Any feasible solution of the partitioning formulation is a feasible solution of the set-covering formulation. Additionally, if there exists a feasible solution in the set-covering formulation that is not feasible for the partitioning problem, it can be transformed into a feasible solution for the partitioning problem with a strictly lower cost $c$ by removing multiple visits to the customers.

The set-covering formulation was introduced to reduce the dual solution space in comparison to the partitioning formulation. While dual variables associated with constraints (2.10) are unconstrained, those associated with constraints (2.13) must be positive.

VRP VARIANTS Numerous real-life applications, frequently originating from logistics and transportation sectors, have been built upon the classical formulation of the VRP. The classical VRP is a relatively simplistic model that does not encompass all the constraints encountered in practical scenarios. Addressing these limitations has been a primary motivation for the scientific community to augment the classical VRP, thereby defining a wide range of variants to better represent real-world situations. These constraints include, but are not limited to:

- problems with asymmetric distances,
- problems with maximum travel times for vehicles,
- problems with time-windows,
- problems with an heterogeneous fleet of vehicles,
- problems with multiple depots,
- periodic problems,
- dynamic problems,
- disrupted problems.

On the other hand, while the original objective of VRP is to minimize the travel cost of the fleet of vehicles, the VRP has been expanded to incorporate a diverse array of objective functions. These objective functions include, but are not limited to:

- minimizing the total time of the routes, from departure to arrival at the depot,
- minimizing the difference between actual arrival time at a customer and the customer's preferred arrival time,
- minimizing the number of vehicles used,
- minimizing greenhouse gas emissions,
- maximizing profits,
- maximizing the difference between profits and costs.

VRP variants can consider one or multiple of the previously presented objectives. Other variants are derived by combining multiple constraints and objectives from the different groups presented above.

### 2.3.2 *Selective Vehicle Routing Problems with profits*

In the TSP, VRP and their variants, all feasible solutions must serve all the customers. However, real-life applications often impose constraints on resources and time, forcing companies to prioritize which customers to serve in order to maximize their profits. This newly introduced class of routing problems is known as Selective VRP with profits. In these problems, each customer is assigned a value, quantifying the significance of serving that particular customer. The VRP with profits follows two simultaneous objectives: firstly, maximizing the total value of served customers by determining which clients to prioritize, and secondly, minimizing travel costs by efficiently scheduling them within the routes of the vehicles. VRP with profits are often encountered in real-life applications. A historically significant application of VRP with profits is the modeling of the sport of orienteering ([Tsiligirides, 1984]). Other applications include airline crew scheduling ([Gopalakrishnan et al., 2005]), mobile healthcare planning ([Doerner et al., 2007]), waste collection ([Han et al., 2015]), collaborative logistics ([Defryn et al., 2016]), etc.

The particular case of VRP with profits that considers only one vehicle is known as the TSP with profits. The problem is similar to the TSP but the vehicle is unable to visit all the customers. Therefore, a solution is a cycle among the cities that starts and ends at the depot, visiting the customers at most once. Three main variants of the TSP with profits have been widely studied, based on their objective function:

- *Orienteering Problem (OP)*. First defined by Tsiligirides (1984), it aims at maximizing the total profit collected from the customers. The length of the route of the vehicle cannot exceed a predefined time limit, introduced as a constraint.

- *Prize Collecting TSP (PCTSP)*. First defined by Balas (1989), it aims at minimizing the total travel costs. The route must visit enough customers to meet a predefined threshold on the profit, introduced as a constraint.

- *Profitable Tour Problem (PTP)*. First defined by Dell'Amico et al. (1995), it aims at optimizing both objectives. The objective function is defined as a linear combination of profit and costs.

The main VRP with profits considering multiple vehicles is the Team Orienteering Problem (TOP). It was first introduced by Butt et al. (1994) as the "Multiple Tour Maximum Collection Problem". A fleet $F$ of $m$ vehicles is available; each vehicle has a maximum travel distance of $T_{max}$. The route of each vehicle starts and ends at the depot, and serves a subset of the $n$ customers, collecting their profit, without exceeding their maximum travel distance. A solution of the TOP must efficiently design the routes of the $m$ vehicles and chose the customers to serve, in order to maximize the total profit collected.

An instance of the TOP can be represented by a graph $G = (V^+, A)$, with $V^+ = V \cup \{0\}$. $V = \{1, 2, ..., n\}$ represents the $n$ customer, and $0$ is the depot. A profit $p_i$ is associated with each customer $i \in V$. Each arc $(i, j) \in A \subseteq V^+ \times V^+$ is associated with a travel cost $t_{ij}$.

The TOP can be modeled as an integer programming model. Binary variables $y_{ir}$ are set to 1 if customer $i$ is served by vehicle $r \in F$, 0 otherwise. Binary variables $x_{ijr}$ are set to 1 if arc $(i,j) \in A$ is used by vehicle $r$, 0 otherwise.

$$\text{Maximize} \sum_{r \in F} \sum_{i \in V} y_{ir} p_i \tag{2.14}$$

$$\sum_{r \in F} y_{ir} \leqslant 1 \quad \forall i \in V \tag{2.15}$$

$$\sum_{i \in V} x_{0ir} = \sum_{i \in V} x_{i0r} \quad \forall r \in F \tag{2.16}$$

$$\sum_{i \in V^+ \setminus k} x_{kir} = \sum_{i \in V^+ \setminus k} x_{ikr} = y_{kr} \quad \forall k \in V,\ r \in F \tag{2.17}$$

$$\sum_{(i,j) \in A} x_{ijr} t_{ij} \leqslant T_{max} \quad \forall r \in F \tag{2.18}$$

$$\sum_{(i,j) \in U \times U} x_{ijr} \leqslant |U| - 1 \quad \forall U \subseteq V,\ |U| \geqslant 2,\ r \in F \tag{2.19}$$

$$x_{ijr} \in \{0,1\} \quad \forall (i,j) \in A,\ r \in F \tag{2.20}$$

$$y_{ir} \in \{0,1\} \quad \forall i \in V,\ r \in F \tag{2.21}$$

Equation (2.14) represents the objective function, that maximizes the total collected profit. Constraints (2.15) ensure that a customer is served by at most one vehicle. Constraints (2.16) and (2.17) are flow constraints. Constraints (2.18) ensure that a vehicle does not travel further than the maximum distance $T_{max}$. Constraints (2.19) are sub-tour elimination constraints. These constraints ensure that the route of every vehicle starts from the depot. Constraints (2.20) and (2.21) define the domain of the decision variables.

Many variants of the TOP have been introduced, adding capacities constraints on the vehicles (Capacitated Team Orienteering Problem (CTOP) [Archetti et al., 2009]) or time window constraints (Team Orienteering Problem with Time Windows (TOPTW) [Amarouche et al., 2020]).

### 2.3.3 *Disruption management*

In all the problems presented in the previous section, the routes are drawn based on instances that are fully defined. However, in real-world applications such instances may not be realistic. Two distinct approaches exist: stochastic optimization and real-time re-planning. Stochastic optimization introduces uncertainties in the values of some parameters, such as travel times or demands. Optimal solutions in stochastic optimization are those that optimize the objective function under the worst-case scenario. Real-time re-planning first computes the routes based on the initial instance, and modifications in the parameters prompt a new computation. These modifications are often seen as disruptions.

Disruption management has been studied in various contexts, including production scheduling, transport planning, supply chain management [Yu et al., 2004], airline scheduling [Clausen et al., 2010], as well as a variety of VRPs [Eglese et al., 2018], such as waste collection [Li et al., 2008] and emergency vehicle routing [Wang et al., 2010]. Yu et al. (2004) defined disruption management as the need to *"dynamically revise the original [optimal or near-optimal] plan"* after disruptions occur due to *"internal or external uncertain factors"*. The new plan must account for the evolving environment and *"[minimize] the negative impact of the disruption"*.

Figure 2.3 shows a standard workflow for replanning for the VRP. Initial routes are computed by solving the standard version of the problem. When a disruption occurs, its impact is evaluated. If the disruption only marginally disturbs the routes, it may be possible to fix the routes with little effort. When the disruption is consequential, new efficient routes are generated by solving the disrupted version of the problem (here, D-VRP). Disruption management can be applied to any variant of the VRP: Single-Depot VRP, TOP and its extensions, VRPTW, CVRP, etc. Several disruptions can be considered, for example: vehicle breakdown, disrupted link in the road network, disruption of supplied goods, changes in customer demands.

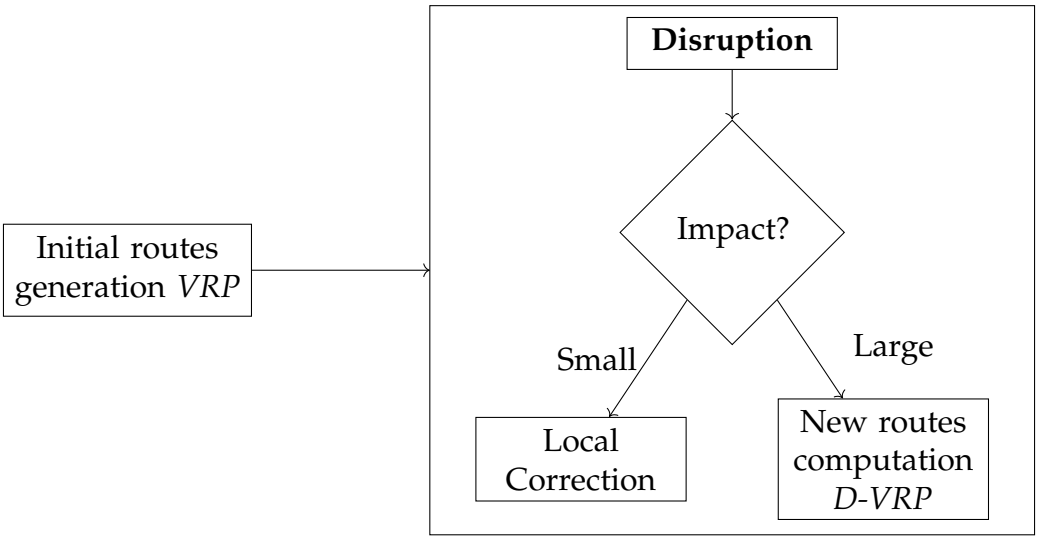



Figure 2.3: Disruption management workflow for the VRP

Disrupted VRP problems differ from their classic counterparts because:

- Typically, in the original problems, the objective is to determine optimal routes for the vehicles that start and end at the depot. When a disruption occurs during the execution of the routes, the vehicles' starting point becomes their current location.

- In contrast to the original problems, where it is acceptable to spend several hours or days obtaining the optimal solution, disruption management requires a quick response. Consequently, disruption management problems are primarily approached through heuristic resolution.

- In disruption management, there is no need to solve the problem from scratch, as it can benefit from the initial solution. This approach is even encouraged to limit deviations from the original plan.
- New constraints may emerge due to changes in conditions or to fulfill commitments from the original plan.
- While the objective of VRP problems is to minimize total operational costs or maximize total collected profit, disruption management problems account for deviations from the original plan. This is achieved by turning the problem into a bi-objective problem or by incorporating a term representing the deviation into the objective function. The additional objectives can represent the costs of deviation (operational costs, new vehicles, cancellation costs, etc.), inconvenience for drivers (number of route changes, waiting times), and/or inconvenience for customers (delays).
- Due to the multiplicity of objectives, the decision-making process may require generating multiple solutions that offer different trade-offs between the objectives, rather than seeking a single "optimal" solution.

## 2.4 BI-OBJECTIVE OPTIMIZATION

In real-world applications, optimization problems may need to simultaneously consider multiple, often conflicting, objectives. In the following, all the objectives are to be minimized, without loss of generality. Any multi-objective problem can be written as:

$$(\text{MOP}) \begin{cases} \text{Minimize } f(x) = (f_1(x), f_2(x), ..., f_p(x)) \\ \text{s.t. } x \in \mathcal{X} \end{cases}$$

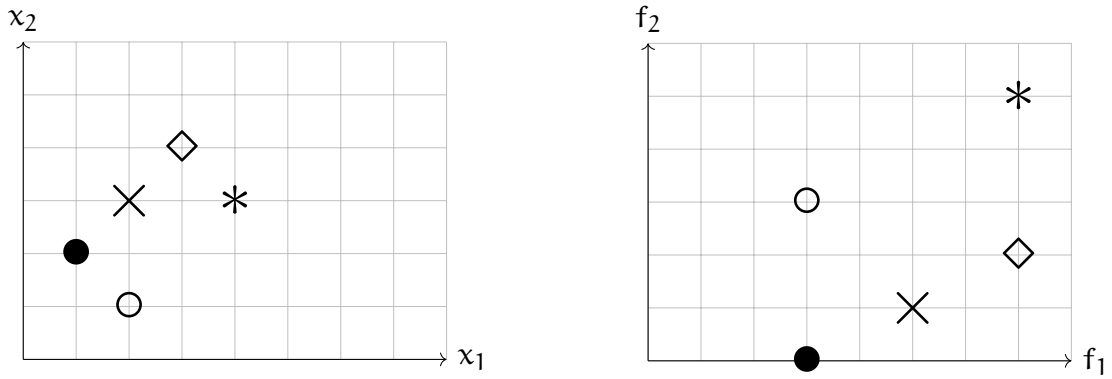
The set $\mathcal{X} \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ is the feasible set, defined over $n_1$ continuous variables and $n_2$ integer variables. The set $\mathcal{Y} = \{f(x) : x \in \mathcal{X}\} \subseteq \mathbb{R}^p$ is the image set. The Euclidean vector space $\mathbb{R}^n$, with $n = n_1 + n_2$, comprising the set of feasible solutions is called solution or decision space. The Euclidean vector space $\mathbb{R}^p$ comprising the image set is called the image or objective space. Figure 2.4 illustrates the link between $\mathcal{X}$ and $\mathcal{Y}$.

**Definition 1** *Consider two solutions* $x, x' \in \mathcal{X}$.

- *x weakly dominates x', denoted by* $x \leqq x' \Leftrightarrow f_i(x) \leqslant f_i(x')\ \forall i \in \{1, 2, ..., p\}$
- *x dominates x', denoted by* $x \leqslant x' \Leftrightarrow \begin{cases} f_i(x) \leqslant f_i(x')\ \forall i \in \{1, 2, ..., p\} \\ f_i(x) < f_i(x')\ \exists i \in \{1, 2, ..., p\} \end{cases}$
- *x stricly dominates x', denoted by* $x < x' \Leftrightarrow f_i(x) < f_i(x')\ \forall i \in \{1, 2, ..., p\}$

**Definition 2** *A solution* $x \in \mathcal{X}$ *is Pareto-optimal (or efficient) if there is no solution* $x' \in \mathcal{X}, x' \neq x$, *such that* $x \leqslant x'$.

*A solution* $x \in \mathcal{X}$ *is weakly Pareto-optimal (or weakly efficient) is there is no solution* $x' \in \mathcal{X}, x' \neq x$, *such that* $x < x'$.




**(a)** Solution Space

**(b)** Objective Space

Figure 2.4: Example of a solution space and the corresponding objective space, with $f_1 = x_1 + x_2$ and $f_2 = 2x_1 - x_2$, $x \in \mathcal{X}$.

**Definition 3** *A point $y \in \mathcal{Y}$ is a non-dominated point if solution $x \in \mathcal{X}$ such that $y = f(x)$ is a Pareto-optimal solution.*

**Definition 4** *A non-dominated point $y \in \mathcal{Y}$ is supported if it is on the convex envelope of $\mathcal{Y}$.*
*A non-dominated point $y \in \mathcal{Y}$ is unsupported if it is inside the convex envelope of $\mathcal{Y}$.*

In the following, solving (MOP) refers to finding the set of all non-dominated points, supported or unsupported, denoted by $\mathcal{Y}_N$, that defines the **Pareto front**. Multiple solutions in $\mathcal{X}$ can be associated with the same point $y \in \mathcal{Y}$. Hence, there are more efficient solutions than the number of non-dominated points. Two sets of non-dominated points are defined:

**Definition 5** *[Hansen, 1980] The minimal complete set of (MOP) is the set of non-dominated points with at least one efficient solution associated with each point.*
*The maximal complete set is the set of non-dominated points with all efficient solutions associated with each point.*

**Definition 6** *A non-dominated supported point $y \in \mathcal{Y}$ is an extreme point if it is an extreme point of the convex envelope of $\mathcal{Y}$.*

**Definition 7** *The extreme point $y \in \mathcal{Y}$ that minimizes objective $f_i$, $i \in \{1, 2, ..., p\}$, is called the $f_i$-extreme point.*

In order to characterize the objective space, we define two specific points that delineate a rectangle where all the non-dominated points are located: the ideal point and the nadir point.

**Definition 8** *The ideal point $y^I$ of $\mathcal{Y}_N$ is defined as $y_i^I = \min_{x \in \mathcal{X}} f_i(x)$.*
*The nadir point $y^N$ of $\mathcal{Y}_N$ is defined as $y_i^N = \max_{x \in \mathcal{X}} f_i(x)$*

In bi-objective optimization, the ideal point is given by the maximum value of each component of the lexicographically optimal images. However, when $p > 2$, there is no efficient algorithm currently known [Ehrgott, 2005a].

We can also delineate a precise search area using local ideal and nadir points.

**Definition 9** *Let* $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$.
*The local ideal point* $\hat{y}^I$ *of* $\hat{\mathcal{Y}}$ *is defined as* $\hat{y}^I_i = \min_{y \in \hat{\mathcal{Y}}} y_i$.
*The local nadir point* $\hat{y}^N$ *of* $\hat{\mathcal{Y}}$ *is defined as* $\hat{y}^N_i = \max_{y \in \hat{\mathcal{Y}}} y_i$

However, in real-world applications, typically only one of these solutions will be chosen. The decision-making process can intervene in different phases:

- *A priori*. The decision-maker is involved in the early stages of the solution process to define preferences between the objectives. These preferences can be defined for each objective independently (e.g., in an aggregated sum of objectives). A priori solution methods only solve one mono-objective problem, but it may be challenging for the decision-maker to precisely define their preferences.

- *Interactive*. The decision-maker is engaged from the beginning of the solution process to define preferences and remains involved throughout to guide the process. Similar to a priori methods, only one solution is returned at the end of the process, but this approach is more adaptable to the decision-maker's needs. The primary drawback of interactive methods is the need for several interventions from the decision-maker, which can be time-consuming.

- *A posteriori*. The decision-maker enters the process only at the end, choosing the most suitable solution from a set of good solutions. These solution methods are more costly, as they require generating multiple solutions. Although a posteriori methods might seem less appealing than the two previous approaches, they provide the decision-maker with insights into all available compromises, enabling them to make an informed decision.

In the following, we will focus on a posteriori solution methods.

### 2.4.1 *Scalarization methods*

In this section, we introduce methods that transform the bi-objective problem into multiple mono-objective problems, which are known as scalarization methods. Below, we list the most common scalarization methods.

WEIGHTED SUM The first method relies on the formulation ($MOP_\lambda$) introduced by Geoffrion (1968), where the objective function is represented as the weighted sum of the objectives.

$$(\mathrm{MOP}_\Lambda) \begin{cases} \text{Minimize } f_\Lambda(x) = \sum_{i=1}^{p} \Lambda_i f_i(x) \\ \text{s.t. } x \in \mathcal{X} \end{cases}$$

An optimal solution of $(\mathrm{MOP}_\Lambda)$ is a (weakly) efficient solution of the problem, given a weight $\Lambda \in \mathbb{R}^p_>$ ($\Lambda \in \mathbb{R}^p_\geqslant$). The non-dominated points are obtained by solving a series of mono-objective problems. The most famous method relying on the weighted sum of objectives is the dichotomic search method [Aneja et al., 1979]. The perpendicular search method [Chalmet et al., 1986] extends the dichotomic search to find the complete set of non-dominated points. The method introduced by Sylva et al. (2004) fixes the weights $\Lambda$ and introduces constraints on the objectives to explore the objective space. Finally, the last method is based on a different scalarization technique using Tchebycheff distances.

DICHOTOMIC SEARCH Introduced by Aneja et al. (1979). This method is guaranteed to identify all extreme non-dominated points and also finds some supported non-extreme points. No unsupported non-dominated points are found. First, the $f_1$ and $f_2$-extreme points $y_1$ and $y_2$ are computed. Then, we iteratively consider each pair of successive non-dominated points $(y, y')$, associated with solutions $x$ and $x'$, in the current objective space. The first pair of points $(y, y')$ considered is $(y_1, y_2)$. The search direction $\hat{\Lambda}$ is computed such that $\hat{\Lambda}(x) = \hat{\Lambda}(x')$, and $(\mathrm{MOP}_{\hat{\Lambda}})$ is solved. If the solution $\hat{x}$ of $(\mathrm{MOP}_{\hat{\Lambda}})$ has a lower cost than both $x$ and $x'$, a new extreme non-dominated point has been found. Otherwise, if the cost of $\hat{x}$ is equal to the cost of either $x$ or $x'$, the non-dominated point associated with $\hat{x}$ is either identical to $y$ or $y'$, or it represents a newly discovered supported non-extreme point.

PERPENDICULAR SEARCH Introduced by Chalmet et al. (1986). The perpendicular search method is similar to the dichotomic search method but aims to find all non-dominated points. During the iterative exploration phase of consecutive non-dominated points $(y, y')$, with associated solutions $x$ and $x'$,where $f_1(x) < f_1(x')$, additional constraints are introduced to restrict the objective space. First, the cost $f_1$ of the solution is constrained to be strictly less than $f_1(x)$. Second, the cost $f_2$ of the solution is constrained to be strictly less than $f_2(x')$.

SYLVA AND CREMA METHOD Introduced by Sylva et al. (2004). This method differs from previous methods in that weights $\Lambda$ are fixed at the beginning of the solution process, satisfying $0 < \Lambda_i < 1$ for all $i \in [\![1, p]\!]$, and $\sum i = 1^p \Lambda_i = 1$. The problem $(\mathrm{MOP}_\Lambda)$ is solved iteratively until no new non-dominated points are found. When a new point is obtained, constraints on the objective values are added to the problem in order to exclude regions dominated by the previously obtained points. As new constraints are introduced, the computational cost of finding solutions increases progressively.

WEIGHTED TCHEBYCHEFF METHOD Introduced by Steuer et al. (1983). This method uses Tchebycheff distances to reference points in order to compute efficient solutions. Changing the reference point in this method has the same effect as changing the weights in the previous methods.

EPSILON-CONSTRAINT Introduced by Haimes (1971). The $\epsilon-constraint$ method does not rely on the weighted sum of objectives. In this method, only one objective is used as the main objective function, while the other objectives are constrained. The constraint values for the objectives are iteratively updated to eliminate the space dominated by points already found. It is based on the following formulation:

$$(MOP_\epsilon) \begin{cases} \text{Minimize } f^\epsilon(x) = f_k(x) \\ \text{s.t. } f_i(x) \leqslant \epsilon_i \text{ , } i \in \{1, 2, ..., p\} \text{ and } i \neq k \\ x \in \mathcal{X} \end{cases}$$

The first step computes the $f_k$-extreme point by setting all $\epsilon_i$ to $+\infty$. The iterative reduction of coefficients $\epsilon_i$ allows to search the entire solution space and find all the non-dominated points.

An optimal solution of $(MOP_\epsilon)$ is a weakly efficient solution of the original problem. Therefore, ensuring that each solution of (MOP$\epsilon$) yields an efficient solution would reduce the number of mono-objective problems to solve. In order to only find efficient solutions, Ozlen et al. (2009) used a weighted objective function to obtain the lexicographical order. When applied to a bi-objective optimization problem, with $f^\epsilon = f_1$, the formulation becomes:

$$(BOP_\epsilon) \begin{cases} \text{Minimize } f(x) = f_1(x) + w_2 f_2(x) \\ \text{s.t. } f_2(x) \leqslant \epsilon \\ x \in \mathcal{X} \end{cases}$$

The weight $w_2$ must be picked wisely: it must be small enough as to not eliminate efficient solutions. Then, among the solutions with the same cost $f^\epsilon$, $(BOP_\epsilon)$ returns the solution that optimizes the other objective. This method can be applied to $p$ objectives, with precisely chosen weights for all $p-1$ secondary objectives.

Other variations that focus on finding only efficient solutions have been proposed. Laumanns et al. (2006) and Hamacher et al. (2007) introduced lexicographical versions of the $\epsilon$-constraint method. Additionally, Ehrgott (2005b) introduced a term in the objective function of $(MOP_\epsilon)$ that penalizes solutions exceeding the secondary objectives. Finally, Mavrotas (2009) introduced an augmented $\epsilon$-constraint method. This approach introduces slack and surplus variables to transform the $\epsilon$ constraints into equality constraints and adds a linear combination of these new variables to the objective function.

For bi-objective VRPs where the resulting mono-objective problem can be solved with column-generation algorithms, Glize et al. (2022) proposed an $\epsilon$-constraint column generation-and-enumeration algorithm.

TWO-PHASES METHOD Two-phase methods are typically approaches that combine two of the previously mentioned methods to compute $\mathcal{Y}_N$. In general, the first phase aims at finding easily obtainable points, narrowing the search space, while the second phase employs a more sophisticated algorithm to find the remaining solutions. For example, Ulungu et al. (1995) first finds all extreme points with a dichotomic search, and then use an $\epsilon$-constraint algorithm for the second phase.

### 2.4.2 *Evaluation*

When heuristic solution methods are implemented to solve (MOP), the heuristic sets of solutions do not only represent non-dominated points. In the following, we present a simple algorithm to test if a solution is efficient. Additionally, it is not as easy to compare two heuristic solutions, unlike mono-objective optimization. One of the most commonly used metrics is the hypervolume, as presented below.

EFFICIENCY OF A SOLUTION Benson's method [Benson, 1978] can be used to verify is a solution $x^0 \in \mathcal{X}$ is efficient.

$$\begin{cases} \text{Maximize } \sum_{i=1}^{p} h_i \\ \text{s.t. } f(x^0) - f(x) - h = 0 \\ \qquad x \in \mathcal{X} \\ \qquad h \in \mathbb{R}^p_+ \end{cases}$$

If $x^0$ is not efficient, this method returns a solution $x^*$ that is efficient.

COMPARISON OF HEURISTIC SOLUTIONS In bi-objective optimization, an upper bound is obtained by solving a relaxation of the problem, and a lower bound is a set of feasible solutions that do not dominate each other. To compare the quality of heuristic methods, one of the most common metrics is hypervolume [Zitzler et al., 1998], or S-volume, which compares the lower bounds obtained. Hypervolume is widely used in multi-objective optimization because it can be computed without knowing the optimal set $\mathcal{Y}_N$. The computation of the hypervolume requires two reference points: an ideal point $y^I$ and a nadir point $y^N$. It is important to use the same reference points when comparing multiple approximate fronts. Let $\Lambda(a_i)$ be the size of the rectangular area $a_i$ constructed with a solution $s_i$ from an approximation set $A$ and the nadir as corners. For approximate set $A$, the hypervolume HV is computed as follows:

$$HV(A) = \frac{\Lambda\left(\bigcup_{a_i \in A} a_i\right)}{(y_1^N - y_1^I)(y_2^N - y_2^I)} \tag{2.22}$$

The hypervolume corresponds to the ratio of the dominated area defined by the approximation set to the total area of the rectangle defined by the ideal and nadir points.

## 2.5 CONCLUSION

In conclusion, this chapter has provided an overview for understanding the decision-making challenges associated with responding to wildfires. Drawing on insights from the GEO-SAFE project, we have explored the domains of fire suppression, life and goods protection, implementation, and training, highlighting their critical roles in effective wildfire response. Furthermore, we have particularly focused on selective vehicle routing

and disruption management, shedding light on how optimization techniques can be applied to enhance response strategies. Finally, we have introduced the concept of bi-objective optimization, necessary to tackle difficult multi-objective decision-making scenarios. This overview sets the stage for the subsequent chapters, where we will delve deeper into a specific optimization problem arising during wildfire response: routing firefighters' crews for asset protection.

# Chapter 3

# Valid inequalities for the Asset Protection Problem

## Contents



### 3.1 Introduction

In this chapter, we will focus on the problem of resource allocation for preventive actions to protect community assets. This problem is referred to as the Asset Protection Problem during an escaped wildfire (APP). The APP was presented as a variant of the Team Orienteering Problem with Time Windows (TOPTW) with a heterogeneous fleet of vehicles and cooperation between the vehicles. Merwe et al. (2015) proposed a MILP formulation to solve the problem. They managed to solve to optimality small-size instances; however instances with 50 community assets to protect were already very difficult to solve. Roozbeh et al. (2018) developed an Adaptative Large Neighborhood Search (ALNS) algorithm based on problem-specific attributes that produces good solutions for large-size instances (up to 200 assets). Nuraiman et al. (2020) proposed a Spatial Decomposition-based Math-heuristic (SDM) approach that outperforms the ALNS algorithm. Yahiaoui et al. (2022) proposed a Greedy Randomized Adaptive Search Procedure with Iterated Local Search (GRASPxILS). The authors used an insertion heuristic based on adaptive candidate lists, a Variable Neighborhood Descent search procedure and a post-optimization phase using a set cover formulation. It yielded better results in lower computational time for almost all the benchmark instances

First, we introduce the problem and its mathematical formulation in Section 3.2. In Section 3.3, we introduce multiple valid inequalities.

- We introduce symmetry-breaking inequalities based on a dominance relations between solutions in Section 3.3.1. These inequalities allow for a better use of the resources by setting lower and upper bounds on the number of vehicles necessary to protect an asset. We introduce the general formulation for the valid inequalities given the bounds, and two MIPs to compute these bounds.
- We introduce the notion of incompatibility/vehicle for assets based on their time window and the travel times of vehicles in Section 3.3.2. We study the graph of incompatibilities to identify cliques of assets to identify subsets of assets that cannot be visited by the same vehicle.
- We further investigate cliques of incompatible assets in Section 3.3.3 by also looking at the resources. We introduce a MIP to compute an upper bound on the number of assets of a clique that can be protected in a solution.

In Section 3.4, we present the benchmark instances and compare the improvements obtained with the different sets of valid inequalities. When adding the valid inequalities, we were able to solve all instances with 35 assets in 10 seconds on average, while only 74 out of the 120 benchmark instances could previously be solved in less than 10 minutes.

## 3.2 PROBLEM DESCRIPTION AND MIP MODEL

During wildfires, community assets such as schools, hospitals, bridges and factories face the risk of being damaged or destroyed. In most cases, this risk can be diminished or nullified if preventive protection actions are taken. Protection requires resources to be dispatched in a timely manner to the asset. These interventions are carried out by the Incident Management Teams (IMT) before the fire reaches the assets. Such actions may include removing fuel materials, wetting down buildings or reducing fire. The time window for performing these actions is crucial: they have to be taken before the fire fronts reach the asset, but not too early for the protection to be efficient. Some interventions may require several trucks with specific capacities, thus requiring different teams to collaborate to perform the task in a synchronous way. Based on fire spread and behavioral models, it is possible to plan routes for each of the teams that take the synchronization and time windows constraints into account, so that a maximum number of community assets are protected.

The first mathematical formulation of the mono-objective APP was proposed by Merwe et al. (2015). An instance of the problem is defined as a directed graph $G = (V, A)$ where $V$ is a finite set of $n$ vertices and $A$ a finite set of arcs.The $n$ vertices represent locations considered in our problem. The first $m$ vertices represent the depots, the $n$-th vertex is the sink node. The remaining $n - m - 1$ vertices are the assets we seek to protect. For convenience, we define three sets $V^d = \{1, ..., m\}$, the set of the depots only; $V^a = \{m + 1, ..., n - 1\}$, the set of the assets only; and $V^- = \{1, ..., n - 1\}$, the set of all locations excluding the sink node. Travel times between the locations satisfy the triangle

inequality. The planning time horizon is continuous. The route of a vehicle is the combination assignment, i.e., the list of assets visited by the vehicle, and the order in which the assets of the assignments are visited.

PROTECTION REQUIREMENT An asset is *protected* in a solution if it contributes to the total protected value, i.e. the vehicles assigned to the asset collectively meet its protection requirements and the starting time of service is within the time window of the asset. Asset protection requirements and vehicle capabilities consist of multiple resources (e.g., crew size, water supply, number of ladders, etc.). Table 3.1 shows an example of ten vehicles and their associated capability vector **cap**, with three resources. For example, an asset $i$ with resource requirement $\mathbf{r}_i = (3, 1, 2)$ can be protected by vehicles 4, 7 and 10 with respective capability vectors $\mathbf{cap}_4 = (2, 1, 0)$, $\mathbf{cap}_7 = (0, 2, 1)$ and $\mathbf{cap}_{10} = (1, 0, 1)$. The starting time of service of an asset must be the same for all the vehicles the asset is assigned to (synchronization). The aim is to generate updated routes that maximize the total value of the protected assets, while minimizing the deviation from the pre-disruption solution.

Table 3.1: Vehicles and their capability vectors

| Vehicle | Resource 1 | Resource 2 | Resource 3 |
|---|---|---|---|
| 1 | 1 | 1 | 2 |
| 2 | 1 | 1 | 2 |
| 3 | 1 | 1 | 2 |
| 4 | 2 | 1 | 0 |
| 5 | 2 | 1 | 0 |
| 6 | 0 | 2 | 1 |
| 7 | 0 | 2 | 1 |
| 8 | 0 | 2 | 1 |
| 9 | 1 | 0 | 1 |
| 10 | 1 | 0 | 1 |

Table 3.2 lists the sets and parameters used in the mathematical formulation of APP. Table 3.3 lists the decision variables and their type.

Table 3.2: Sets and parameters used in the mathematical formulation of APP

| Symbol | Definition |
|---|---|
| $n$ | Number of vertices in the graph representation of the problem |
| $m$ | Number of depots |
| $V$ | Set of locations |
| $V^a$ | Set of locations limited to the assets to protect |
| $V^d$ | Set of locations limited to the depots |
| $V^-$ | Set of all locations except the sink node $n$ |
| $Q$ | Set of possible vehicle types |
| $p_q$ | Number of vehicles of type $q$ available |
| $\mathcal{E}_q$ | Set of available arcs for vehicles of type $q$ |
| $\delta_q^+(i)$ | Set of locations directly reachable from location $i$ by a vehicle of type $q$ |
| $\delta_q^-(i)$ | Set of locations from where location $i$ can be directly reached by a vehicle of type $q$ |
| $a_i$ | Service duration associated with asset $i$ |
| $o_i$ | Earliest start of service at asset $i$ |
| $c_i$ | Latest start of service at asset $i$ |
| $\mathbf{r}_i$ | Protection requirement vector of asset $i$ |
| $v_i$ | Value of asset $i$ |
| $t_{ijq}$ | Travel time from asset $i$ to asset $j$ of a vehicle of type $q$ |
| $\mathbf{cap}_q$ | Capability vector associated with a vehicle of type $q$ |
| $stock_{iq}$ | Number of vehicles of type $q$ initially available at depot $i$ |

Table 3.3: Decision variables used in the mathematical formulation of APP

| Symbol | Definition |
|---|---|
| $X_{ijq}$ | integer, the number of vehicles of type $q$ traveling directly from asset $i$ to asset $j$ |
| $Zijq$ | binary, 1 if at least one vehicle of type $q$ travels directly from asset $i$ to asset $j$ |
| $Y_i$ | binary, 1 if asset $i$ is protected, 0 otherwise |
| $S_i$ | continuous, the start time of service at asset $i$ |

The mathematical model, hereafter denoted by (APP), is written as follows:

$$Maximize\ f_1 = \sum_{i \in V^a} v_i Y_i \tag{3.1}$$

$$\sum_{j \in \delta_q^+(i)} X_{ijq} = stock_{iq} \quad \forall i \in V^d,\ q \in Q \tag{3.2}$$

$$\sum_{i \in \delta_q^-(k)} X_{ikq} = \sum_{j \in \delta_q^+(k)} X_{kjq} \quad \forall k \in V^a,\ q \in Q \tag{3.3}$$

$$X_{ijq} \leqslant p_q Z_{ijq} \quad \forall (i,j) \in \mathcal{E}_q,\ q \in Q \tag{3.4}$$

$$\sum_{q \in Q} \sum_{i \in \delta_q^-(k)} X_{ikq} \mathbf{cap}_q \geqslant Y_k \mathbf{r}_k \quad \forall k \in V^a \tag{3.5}$$

$$S_i + t_{ijq} + a_i \leqslant S_j + M(1 - Z_{ijq}) \quad \forall (i,j) \in \mathcal{E}_q,\ q \in Q \tag{3.6}$$

$$S_i \geqslant o_i \quad \forall i \in V^a \tag{3.7}$$

$$S_i \leqslant c_i \quad \forall i \in V^a \tag{3.8}$$

$$Y_i \in \{0,1\} \quad \forall i \in V^a \tag{3.9}$$

$$X_{ijp} \in \{0,1,2,...p_q\},\ Z_{ijq} \in \{0,1\} \quad \forall (i,j) \in \mathcal{E}_q,\ q \in Q \tag{3.10}$$

$$S_i \in \mathbb{R} \quad \forall i \in V \tag{3.11}$$

Equation (3.1) represent the objective function $f_1$, the total protected value.

Constraints (3.2)-(3.4) are the flow constraints for each type of vehicles. Constraints (3.5) ensure that an asset can be protected only if the vehicles assigned to the asset collectively meet the requirements on each resource. Constraints (3.6) ensure time integrity when visiting two assets consecutively. If asset $j$ is directly visited after asset $i$ by a vehicle $p$, service at asset $j$ should not start before service at asset $i$ is over and vehicle $p$ has had enough time to travel from asset $i$ to asset $j$. Constraints (3.7) and (3.8) ensure that the start time of service of any protected asset is within its time window.

Constraints (3.9)-(3.11) define the domain of the decision variables.

## 3.3 VALID INEQUALITIES

In this section, we propose sets of valid inequalities for the APP. We studied the structure of (APP) deduce valid inequalities that strengthen the model, improving the linear relaxation and reducing the solution time. New valid inequalities have been introduced to solve orienteering problems. Fischetti et al. (1998) proposed new families of valid inequalities for the Orienteering Problem (OP). Perboli et al. (2018) did a similar study for the Two-Echelon Capacitated Vehicle Routing Problem (2E-CVRP). El-Hajj et al. (2016) presented strong clique cuts deduced from incompatibility graphs for the Team Orienteering Problem (TOP).

### 3.3.1 *Bounds on protection*

In this section, we introduce valid inequalities based on bounds on the number of vehicles required to protect an asset given its resource requirements. The valid inequalities rely on the following dominance property on the solutions of (APP):

**Proposition 1** *For any solution $\Psi$ of (APP), there exists a solution with the same total protected value such that all assets visited by at least one vehicle are protected.*

**Proof 1** *Consider a solution $\Psi$ of (APP) where an asset $i$ is visited by a vehicle $p$ but asset $i$ is not protected. Asset $i$ is not protected, thus do not participate in total protected value. Removing asset $i$ from the route of vehicle $p$ does not impact the time of visit of the remaining assets on the route, due to the triangle identity, and asset $i$ is still protected because vehicle $p$ was redundant. The solution obtained by removing non-protected assets from the routes of solution $\Psi$ is a feasible solution for (APP) and strictly protects the same assets as $\Psi$, hence having the same total protected value.* ■

Proposition 1 reduces the feasible space we need to explore to find the optimal solution. There is an optimal solution solution of (APP) such that the following inequalities are valid:

$$Y_i \geqslant Z_{ijq} \quad \forall (i,j) \in \mathcal{E}_q,\ q \in \mathcal{Q} \tag{3.12}$$

If a vehicle of type $q$ uses arc $(i,j)$, then asset $i$ is necessarily protected. If asset $i$ is not protected, then no vehicle of type $q$ can use arc $(i,j)$.

*Lower bound on vehicle additions for protection*

The protection of an asset depends on multiple vehicles assigned to the asset to meet its resource requirement. Let $lb_v(i)$ be a lower bound on the number of vehicles required to protect asset $i$. If an asset $i$ is not visited by at least $lb_v(i)$ vehicles, it cannot be protected. For example, for an asset with a requirement vector $r_i = (3,3,3)$, there is no combination of two vehicles from Table 3.1 such that each resource requirement is met. However, vehicles 1, 4 and 6 have a cumulative capability vector of $(3,4,3)$, thus covering the resource requirement of the asset. For such an asset, we have $lb_v(i) = 3$.

Given Proposition 1, there is an optimal solution of (APP) where if asset $i$ is not protected there is no vehicle visiting the asset, otherwise asset $i$ is visited by at least $lb_v(i)$ vehicles. The following set of inequalities is therefore valid:

$$\sum_{q \in \mathcal{Q}} \sum_{k \in \delta^-(i)} X_{kiq} \geqslant lb_v(i) Y_i \quad \forall i \in V^a \tag{3.13}$$

COMPUTATION OF $lb_v(i)$ We introduce a MIP to compute $lb_v(i)$ for a given asset $i$. For each type of vehicles $q \in \mathcal{Q}$, there are $p_q$ vehicles available. For each type of vehicle $q \in \mathcal{Q}$, we introduce an integer variable $W_q$ that represent the number of vehicles of type

$q$ used to protect asset $i$. We aim at minimizing the number of vehicles used to cover the resource requirement $\mathbf{r}_i$ of asset $i$.

$$\text{Minimize} \sum_{q \in Q} W_q \tag{3.14}$$

$$\sum_{q \in Q} W_q \mathbf{cap}_q \geqslant \mathbf{r}_i \tag{3.15}$$

$$W_q \in \{0, 1, 2, ..., p_q\} \quad \forall q \in Q \tag{3.16}$$

Objective (3.14) minimizes the number of vehicles assigned to the asset. Constraint (3.15) ensures that the cumulative capability vector of the selected vehicles meets the resource requirement of the asset. Constraints (3.16) define the domain of the decision variables.

*Upper bound on vehicle additions for protection*

Valid inequalities (3.13) offer a lower bound on the deviation induced by adding an asset to the initial routes of vehicles in order to protect it. We introduce the notion of *minimal protection* of an asset.

**Definition 10** *A set of vehicles represents a* minimal protection *for an asset if the vehicles meet the resource requirement of the asset but no subset of the vehicles does.*

In other words, removing any vehicle from the set makes at least one of the resources of the cumulative capability vector strictly inferior to the requirement of the asset. For example, for an asset with a requirement vector $r_i = (3, 3, 3)$, vehicles 1, 4 and 6 provide minimal protection of the asset. However, vehicles 4, 5, 6, 7, 9 with cumulative capability vector $(5, 6, 3)$ protect asset $i$ even when removing vehicle 5: the protection is not minimal. We say that vehicle 5 is *redundant*. We deduce the following dominance property on the solutions of (APP):

**Proposition 2** *For any solution $\Psi$ of (APP), there exists a solution with the same total protected value such that the vehicles assigned to any asset represent a minimal protection of the asset.*

**Proof 2** *Consider a solution $\Psi$ where an asset $i$ is protected by a set of vehicles that do not represent a minimal protection for the asset. There is at least one vehicle $p$ assigned to asset $i$ that is redundant. Removing asset $i$ from the route of vehicle $p$ does not impact the time of visit of the remaining assets on the route, due to the triangle identity, and asset $i$ is still protected because vehicle $p$ was redundant. The solution obtained by removing assets from the routes of redundant vehicles is a feasible solution for (APP) and strictly protects the same assets as $\Psi$, hence having the same total protected value.* ■

Let $ub_v(i)$ be an upper bound on the number of vehicles that can be assigned to asset $i$ without any of these vehicles being redundant. There is an optimal solution of (APP)

such that if asset $i$ is not protected, no vehicles visit asset $i$ (Proposition 1), otherwise at most $ub_v(i)$ are assigned to the protection of asset $i$ (Proposition 2). The following set of valid inequalities thus holds:

$$\sum_{q\in Q}\sum_{k\in\delta^-(i)} X_{kiq} \leqslant ub_v(i)Y_i \quad \forall i \in V^a \tag{3.17}$$

COMPUTATION OF $ub_v(i)$ We introduce a MIP to compute $ub_v(i)$ for a given asset $i$. We will denote $\mathcal{P}$ the set of available vehicles, representing the $p_q$ vehicles of each type $q \in Q$. We note $cap_{pu}$ the $u$-th resource of capability vector $\mathbf{cap}_p$ and $r_{iu}$ the $u$-th resource of requirement vector $r_i$. For each vehicle $p \in \mathcal{P}$, a binary decision variable $W_p$ is set to 1 if vehicle $p$ is assigned to the protection of the asset. For each vehicle $p \in \mathcal{P}$ and each resource $u$, a binary decision variable $Y_{pu}$ is equal to 1 if resource $u$ is still covered when removing vehicle $p$. We aim at finding the maximum number of vehicles in $\mathcal{P}$ needed to cover the resource requirement of the asset without any of the vehicles being redundant.

$$Max \sum_{p\in\mathcal{P}} W_p \tag{3.18}$$

$$\sum_{p\in\mathcal{P}} W_p \mathbf{cap}_p \geqslant \mathbf{r}_i \tag{3.19}$$

$$\sum_{p'\in\mathcal{P}\setminus\{p\}} W_{p'} cap_{p'u} \leqslant r_{iu} - 1 + M_{pu}Y_{pu} \quad \forall u = 1,...,K,\ p \in \mathcal{P} \tag{3.20}$$

$$\sum_{u=1}^{K} Y_{pu} \leqslant K - W_p \quad \forall p \in \mathcal{P} \tag{3.21}$$

$$W_p \in \{0,1\} \quad \forall p \in \mathcal{P} \tag{3.22}$$

$$Y_{pu} \in \{0,1\} \quad \forall u = 1,...,K,\ p \in \mathcal{P} \tag{3.23}$$

Objective (3.18) maximizes the number of vehicles assigned to the asset.

Constraint (3.19) ensures that the cumulative capability vector of the selected vehicles covers the resource requirement of the asset. Constraints (3.20) and Constraints (3.21) ensure that if a vehicle $p \in \mathcal{P}$ is selected, at least one resource is no longer covered when vehicle $p$ is removed. In other words, if a vehicle is selected, it is not redundant. To ensure that constraints (3.20) are verified when the $u$-th resource is still covered or when vehicle $p$ is not selected, we define:

$$M_{pu} = 1 - r_{pu} + \sum_{p'\in\mathcal{P}\setminus\{p\}} cap_{p'u}$$

Constraints (3.22) and (3.23) define the domain of the decision variables.

### 3.3.2 *Incompatibility/vehicle clique cuts*

In Section 3.3.1, we focused on valid inequalities derived from the resource requirement of a specific asset. In this section, we present two sets of valid inequalities based on incompatibility between assets depending on their time windows.

Two assets $i$ and $j$ are incompatible for a vehicle type $q$ if the same vehicle of type $q$ cannot visit assets $i$ and $j$ within their respective time window.

Let $G_q^{inc/v} = (V^a, E_q^{inc/v})$ be the graph of incompatibilities for vehicle type $q$ between assets where:

$$E_q^{inc/v} = \{(i,j) \mid i,j \in V^a \ : \ o_i + a_i + t_{ijq} > c_j \ \wedge \ \ o_j + a_j + t_{jiq} > c_i\}$$

A clique is a subset of nodes in an undirected graph that are pairwise adjacent. Thus, assigning a vehicle to a protected asset that is part of a clique of the incompatibility graph $G_q^{inc/v}$ excludes all other protected assets of the clique from being visited by this vehicle.

Consider a clique $\mathcal{C}$ of graph $G_q^{inc/v}$. According to Proposition 1, there is an optimal solution of (APP) where a single vehicle of type $q$ can only visit at most one asset of clique $\mathcal{C}$. The following inequality is thus valid for any clique $\mathcal{C}$:

$$\sum_{j \in \mathcal{C}} \sum_{i \in \delta_q^-(j)} x_{ijq} \leqslant p_q \tag{3.24}$$

### 3.3.3 *Clique protection cuts*

In Sections 3.3.1 and 3.3.2, we presented valid inequalities based on resource requirements or incompatibility, respectively, between the assets. In this section, we present a new set of valid inequalities that rely on resource requirements within a clique of incompatible assets.

Two assets $i$ and $j$ are *strictly incompatible* if the assets are incompatible for all vehicle types. Let $G^{inc/s} = (V_a, E^{inc/s})$ be the graph of strict incompatibilities between assets where:

$$E^{inc/s} = \{(i,j) \mid (i,j) \in E_q^{inc/v} \ \forall q \in Q\}$$

A clique $\mathcal{C}$ of graph $G^{inc/s}$ is also a clique in the incompatibility graph $G_q^{inc/v}$ for any vehicle type $q$. Each vehicle can then only be assigned to protect at most one of the assets of the clique. Let $ub(\mathcal{C})$ be an upper bound on the maximum number of assets of a clique $\mathcal{C}$ for which resource requirements are met, with every available vehicle being assigned to at most one asset of the clique. In any solution, at most $ub(\mathcal{C})$ assets of clique $\mathcal{C}$ can be protected. The following inequality is valid for any clique $\mathcal{C}$:

$$\sum_{i \in \mathcal{C}} y_i \leqslant ub(\mathcal{C}) \tag{3.25}$$

COMPUTATION OF $ub(\mathcal{C})$ We introduce a MIP to compute $ub(\mathcal{C})$ for any given clique $\mathcal{C}$ of the incompatibility graph $G^{inc/s}$. For each type of vehicles $q \in \mathcal{Q}$, there are $p_q$ vehicle savailable. For each asset $i$ in the clique, an integer decision variable $W_{iq}$ represent the number of vehicles of type $q$ assigned to asset $i$. A binary decision variable $\beta_i$ is set to 1 if the cumulative capability vector of vehicles assigned to asset $i$ meets the resource requirement of the asset. We aim at maximizing the number of assets of the clique that have their resource requirement met, using each vehicle at most once.

$$\text{Maximize} \sum_{i \in \mathcal{C}} \beta_i \tag{3.26}$$

$$\sum_{q \in \mathcal{Q}} W_{iq} \mathbf{cap}_q \geqslant \beta_i \mathbf{r}_i \quad \forall i \in \mathcal{C} \tag{3.27}$$

$$\sum_{i \in \mathcal{C}} W_{iq} \leqslant p_q \quad \forall q \in \mathcal{Q} \tag{3.28}$$

$$W_{iq} \in \{0, 1, 2, ..., p_q\} \quad \forall i \in \mathcal{C},\ q \in \mathcal{Q} \tag{3.29}$$

$$\beta_i \in \{0, 1\} \quad \forall i \in \mathcal{C} \tag{3.30}$$

Objective (3.26) maximizes the number of assets that have their resource requirement met. Constraints (3.27) ensure that if an asset is protected, the cumulative capability vector of the vehicles assigned to the asset meets the resource requirement. Constraints (3.28) ensure that at most $p_q$ vehicles of type $q$ are used across the clique. Constraints (3.29) and (3.30) define the domain of the decision variables.

## 3.4 COMPUTATIONAL EXPERIMENTS

In this section, we compare the results obtained with the valid inequalities to those obtained on the extended Solomon instances, as used by Roozbeh et al. (2018). There are 60 instances with 200 assets, divided into R1, C1, RC1, R2, C2 and RC2 classes based on their spatial distribution within the 140km by 140km grid. Each class is composed of 10 instances. Solomon instances with less than 200 assets are created using a subset of the original instances. Time windows have a fixed length for every asset. The vehicles are initially available in a central depot. We consider the number of resources $K = 3$ and three types of vehicles with respective capability vectors (1,0,0), (0,1,0) and (0,0,1).

We carried out computational testing on a computer with an Intel Core i7-8550U processor and 8GB of RAM. We implemented the models in Julia, and solved the problems using CPLEX 12.10.

In Section 3.3, we presented valid inequalities adapted to the generation of initial routes. We can divide the valid inequalities for APP in three groups:

- Deviation-based inequalities (DE): (3.12) + (3.13) + (3.17)
- Incompatibility/vehicle clique inequalities (IC): (3.24)
- Clique protection inequalities (PR): (3.25)

We solved the APP for the Solomon instances with 35 assets using the formulation proposed by Merwe et al. (2015) with different combinations of the valid inequalities. We consider three types of vehicles, and two different sets of vehicles Set1 = ($p_1 = 4$, $p_2 = 3$, $p_3 = 2$) and Set2 = ($p_1 = 5$, $p_2 = 4$, $p_3 = 3$). We set a time limit of 600 seconds.

Table 3.4 shows the average solution time for the different classes of Solomon instances and sets of vehicles. The following columns show the results based on which valid inequalities have been added. We show in Columns "ST" the average solution time for instances solved to optimality and in Columns "#Opt" the number of solutions proven optimal within the time limit.

Table 3.4: Computational results for APP for Solomon instances with 35 assets

| Class | Vehicles | None | | (DE)+(IC) | | (DE)+(PR) | | (IC)+(PR) | | All | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | #Opt | ST | #Opt | ST | #Opt | ST | #Opt | ST | #Opt |
| C1 | Set1 | 28.10 | 3 | 9.34 | 10 | 10.37 | 10 | 8.39 | 2 | 9.7 | 10 |
| | Set2 | 78.7 | 8 | 7.56 | 10 | 6.57 | 10 | 103.46 | 9 | 7.49 | 10 |
| C2 | Set1 | – | 0 | 24.68 | 10 | 18.78 | 10 | – | 0 | 15.44 | 10 |
| | Set2 | – | 0 | 32.42 | 10 | 29.76 | 10 | – | 0 | 29.14 | 10 |
| R1 | Set1 | 83.75 | 9 | 5.66 | 10 | 4.66 | 10 | 53.81 | 9 | 4.55 | 10 |
| | Set2 | 81.17 | 10 | 6.18 | 10 | 5.96 | 10 | 61.9 | 10 | 5.68 | 10 |
| R2 | Set1 | 90.22 | 6 | 7.26 | 10 | 7.2 | 10 | 33.36 | 5 | 6.73 | 10 |
| | Set2 | 33.8 | 10 | 3.63 | 10 | 3.3 | 10 | 18.05 | 9 | 3.69 | 10 |
| RC1 | Set1 | 135.48 | 6 | 10.17 | 10 | 7.84 | 10 | 140.27 | 8 | 8.24 | 10 |
| | Set2 | 32.36 | 9 | 4.56 | 10 | 6.23 | 10 | 26.98 | 9 | 4.7 | 10 |
| RC2 | Set1 | 53.28 | 3 | 7.79 | 10 | 6.61 | 10 | 54.83 | 5 | 6.11 | 10 |
| | Set2 | 26.87 | 10 | 2.73 | 10 | 2.3 | 10 | 16.49 | 9 | 2.72 | 10 |
| All | Set1 | 65.14 | 27 | 10.82 | 60 | 9.24 | 60 | 48.44 | 29 | 8.46 | 60 |
| | Set2 | 42.15 | 47 | 9.51 | 60 | 9.02 | 60 | 37.81 | 46 | 8.9 | 60 |

From Table 3.4, we see that we drastically reduce the solution time when we add the deviation-based inequalities (DE). Inequalities (IC) and (PR) help further improve the gains obtained by inequalities (DE). Without valid inequalities, we could only solve 27 instances with 9 vehicles within the time limit, and 47 instances with 12 vehicles. We solved all instances from all classes to optimality in less than 9 seconds on average, with a maximum of 30 seconds for the C2 instances with the second set of vehicles.

As expected with large instances, we could not overcome the limitations of MIP resolution for instances with 100 assets. Our results were inconclusive. After 30 minutes, the solver returned low-quality feasible solutions, and adding the valid inequalities only marginally improved root relaxation. State-of-the-art heuristic solution methods, such as Adaptive Large Neighborhood Search (ALNS) [Roozbeh et al., 2018], Spatial Decomposition based Math-heuristic (SDM) [Nuraiman et al., 2020], or Greedy Randomized Adaptive Search Procedure coupled with an Iterated Local Search (GRASP-ILS) [Yahiaoui et al., 2022], remain far more effective than MIP resolution.

## 3.5 CONCLUSION

In this chapter, we introduced a novel selective vehicle routing problem: the Asset Protection Problem during escaped wildfires (APP). The APP is a variant of the Team Orienteering Problem with time windows and synchronization constraints, involving multiple vehicles that collaborate to protect community assets from fire damage.

Within the existing literature, a Mixed-Integer Programming (MIP) model was proposed to tackle the APP. However, solving this MIP required extensive computational time, even for relatively small instances, making it an unreliable solution method. Consequently, prior research efforts primarily focused on heuristic approaches, which were capable of providing good approximate solutions in significantly shorter time.

Our contribution involved a study of the problem's structure, enabling us to introduce three sets of valid inequalities to enhance the model and reduce solution times. The first set of inequalities aimed to establish bounds on the number of vehicles visiting assets based on resource requirements. The second set of inequalities used the graph structure of asset distribution to identify cliques of incompatible assets according to distances and time windows. The last set combined incompatibility properties and resource sharing to limit the number of assets that can be protected in certain subsets.

The incorporation of these valid inequalities into the MIP formulation allowed us to significantly reduce solution times for all 35-asset instances from hours to just a few seconds on average. It constitutes an encouraging first step in generating optimal solutions to the APP. These results motivate us to extend these valid inequalities to the disrupted version of the APP, as presented in Chapter 4.

# Chapter 4

# Disrupted Asset Protection Problem

## Contents



### 4.1 Introduction

In Chapter 3, we introduced the Asset Protection Problem (APP) which aims at routing a fleet of vehicles to carry out preventive protection operations to a set of community assets in the event of a wildfire. Unfortunately, due to the destructive and uncertain nature of wildfires, disruptions may occur and invalidate the planned routes. The routes need to be updated to consider the effects of these disruptions. Possible disruptions include, but are not limited to:

- vehicle breakdowns. People and vehicles suffer from the difficult conditions in which they operate. It may then be necessary to reroute other vehicles to make up for the breakdown.

- changes in the weather. Wildfire spread is highly sensitive to weather changes. An unpredicted change in wind speed or direction may impact the timing of the operations.

- changes in driving conditions. A road may be blocked by a fallen tree, jammed by people evacuating their homes, or reserved for emergency vehicles only. These changes impact the estimated travel times between the assets.

- delays in operations. Incidents such as a water supply problem may increase the time necessary to carry out the operation protection.

It is then crucial to update the initial routes and reallocate the resources in response to these unexpected changes. Delays and fire propagation changes may make vehicles arrive after the closing of the time window, and breakdowns may create a shortage of required resources. The closing of certain roads may even make an asset unreachable and completely invalidate the initial routes. The problem at-hands is the static problem that arises when a disruption occurs, and could be included in a framework that initially solves APP and handles dynamic disruptions.

Merwe et al. (2017) introduced D-APP and adapted the MIP formulation of [Merwe et al., 2015] to include the measurement of the deviation from an initial assignment. This is a bi-objective problem where total protected value must be maximized while minimizing the deviation from the initial routes. To date, this is the only other study that exists on the dynamic rerouting of the APP.

The D-APP is a new extension of the VRP with synchronization constraints and a dynamic component. Drexl (2012) offered a survey of the different synchronization constraints related to the VRP. Problems with exact operation synchronization have been solved through column generation or branch-and-price, but required either an involved labelling algorithm [Bélanger et al., 2006] [Ioachim et al., 1999], or the reformulation of the problem [Dohn et al., 2009]. Heuristics methods are complicated to implement due to the temporal interdependencies between vehicles. Most of the promising heuristics methods include indirect searches [Li et al., 2005] or constraint programming [Hachemi et al., 2011].

The dynamic component of the D-APP comes from the need to reschedule the routes of the vehicles after a disruption occurs. In their review, Eglese et al. (2018) have summarized the characteristics of disruption management for VRPs. After an initial plan was drawn, disruptions occurred on the customers and/or the vehicles and the plan needs substantial revision. The new plan must take into account the deviation from the initial plan, either by introducing new costs related to changes in the routes (operational costs, cancellation costs, new vehicles costs) or a secondary objective (number of changes in the routes, sum of time delays). Papers on disruption management have often focused on real-world case studies, where time for replanning is limited. Hence, the authors have mostly proposed heuristic solution methods. Mu et al. (2011) introduced the disrupted VRP in which the VRP plan needs to be updated after a vehicle breaks down. The authors presented two Tabu Search algorithms to quickly compute a new routing solution. Li et al. (2009) solved a disrupted VRP with time windows using a Lagrangian

relaxation-based heuristic. However, exact solution methods could be used in order to evaluate the quality of the strategies retroactively.

In this chapter, we focus on the exact solution of D-APP using a MIP. We introduce the D-APP and its mathematical formulation in Section 4.2. In Section 4.3, we present a new mathematical formulation for our problem based on a dominance property on the solutions. In Section 4.4, we extend the three groups of valid inequalities we proposed for APP in Chapter 3 to the disrupted version of the problem. Computational testing in Section 4.5 shows that our reformulation and valid inequalities can be used to solve instances of larger size than the initial model presented by Merwe et al. (2017). Adding the valid inequalities to the initial formulation allowed us to more reliably solve medium-sized instances. The reformulation by itself outperformed the initial formulation with valid inequalities, with solution times decreasing for each of our benchmark instances. The use of valid inequalities with the reformulation additionally reduced the solution time for 60-asset instances by an average of 55 seconds.

## 4.2 PROBLEM DESCRIPTION AND MODEL

When a disruption occurs, the current routes of the vehicles may become obsolete and, consequently, need to be updated. Altering the route of a vehicle too much from the original plan may cause problems. At the time they are dispatched, vehicles have the information about the routes, the assets they are going to protect, and may require specialized equipment. Communicating in such a context is difficult: limiting the number of changes on the routes can alleviate the problem. When updating the routes, the primary objective remains to maximize the total value of protected assets. The secondary objective is to minimize the deviation from the initial plans.

Contrary to mono-objective optimization, we do not aim at generating the optimal solution but, instead, a set of solutions that describe the optimal trade-off surface referred to as Pareto front. A solution is part of the Pareto front if it is non-dominated (or Pareto-optimal), i.e., no other feasible solution to our problem strictly improves one of the objectives without degrading the other. Without additional input about preferences from the decision-maker, all solutions on the Pareto front are of equal interest.

An instance of D-APP is similar to an instance of APP (a set of locations that have a graph representation G and a set of vehicles) but adds a new parameter: an initial solution representing the routes of the vehicles before the disruption. The notations we use to define the mathematical formulation of D-APP are largely similar to those we used for APP in Section 3.2. Table 4.1 highlights the sets and parameters that differ from APP (Table 3.2). The main changes come from the need to consider each vehicle separately: the vehicles of the same type are no longer interchangeable as they have different initial routes.

DISRUPTION The pre-disruption solution $\Phi$ defines the initial routes for the available vehicles. We make no assumptions about the quality of solution $\Phi$ before the disruption. We do not know how nor which parameters have been modified by the disruption. The vehicles assigned to asset $i$ in $\Phi$ are listed in $\mathcal{P}_i^{\Phi}$, a subset of the available vehicles $\mathcal{P}$.

Table 4.1: Sets and parameters updated for the mathematical formulation of D-APP

| Symbol | Definition |
|---|---|
| $\mathcal{P}$ | Set of available vehicles after the disruption |
| $\mathcal{E}_p$ | Set of available arcs for vehicle $p$ |
| $\delta_p^+(i)$ | Set of locations directly reachable from location $i$ by vehicle $p$ |
| $\delta_p^-(i)$ | Set of locations from where location $i$ can be directly reached by vehicle $p$ |
| $t_{ijp}$ | Travel time from asset $i$ to asset $j$ of vehicle $p$ |
| $\mathbf{cap}_p$ | Capability vector associated with vehicle $p$ |
| $start_{ip}$ | 1 if vehicle $p$ starts at depot $i$, 0 otherwise |

Table 4.2: Sets and parameters used in the mathematical formulation of D-APP only

| Symbol | Definition |
|---|---|
| $\Phi$ | Pre-disruption solution |
| $\mathcal{U}_p^\Phi$ | Set of assets not assigned to vehicle $p$ in $\Phi$ |
| $\mathcal{V}_p^\Phi$ | Set of assets assigned to vehicle $p$ in $\Phi$ |
| $\mathcal{P}_i^\Phi$ | Subset of the vehicles $\mathcal{P}$ assigned to asset $i$ in $\Phi$ |
| $\overline{\mathcal{P}_i^\Phi}$ | Subset of the vehicles $\mathcal{P}$ not assigned to asset $i$ in $\Phi$ |
| $x_{ijp}$ | 1 if vehicle $p$ uses arc $(i,j)$ in $\Phi$, 0 otherwise |

Table 4.2 present the new sets and parameters used in the mathematical formulation for D-APP related to the initial solution $\Phi$.

DEVIATION MEASUREMENT We consider the general deviation measurement introduced by Merwe et al. (2017) with unitary costs. The deviation is thus represented by the number of asset/vehicle reassignments: each asset added to or removed from the initial route of a vehicle induces a deviation of one. The aim is to generate updated routes that maximize the total value of the protected assets, while minimizing the deviation from the pre-disruption solution.

We define two properties for the arcs of our problem: valid arcs and out-of-window arcs.

**Definition 11** *The subset of arcs* $\mathcal{E}_p^v = \{(i,j) \mid o_i + t_{ijp} + a_i \leqslant c_j\}$ *is called the set of* valid arcs *for vehicle* $p$. *An arc* $(i,j)$ *is valid for vehicle* $p$ if by starting the service at asset $i$ at the opening of its time window, traveling to asset $j$ after serving asset $i$ does not exceed the closing time of the time window of asset $j$.

**Definition 12** *The subset of arcs* $\mathcal{E}_p^o = \{(i,j) \mid i \in V^a,\ j \in \mathcal{V}_p^\Phi\}$ *is called the set of* out-of-window arcs *for vehicle* $p$. *It represents all the possible arcs entering the assets assigned to vehicle* $p$ *in the pre-disruption solution* $\Phi$.

An arc between two assets $i$ and $j$ can be, for a given vehicle $p$:

- Valid only, in which case the arc can be used to visit asset $j$ after asset $i$ when the two assets are visited within their time windows;

- Out-of-window only, in which case the arc can be used to visit asset $j$, possibly outside of its time window, after asset $i$ when asset $j$ is not protected;
- Valid and out-of-window, in which case the arc can be used in the two previous configurations;
- None, in which case the arc will not be used.

Valid arcs are based on the time windows of the assets: they are necessary and sufficient for the visits between protected assets that must be visited within their respective time windows. Out-of-window arcs are necessary for visiting the assets outside of their time window. An asset visited outside of its time window can be placed anywhere in the route of vehicle $p$; consequently, the vehicle must be able to reach it from any location. Out-of-window arcs are only defined for the assets visited by the vehicle in the pre-disruption solution. We can show that these are the only assets that can be visited by vehicle $p$ outside of their time windows.

**Proposition 3** *In a non-dominated solution, if an asset $i$ is unprotected, only vehicles that are assigned to asset $i$ in the pre-disruption solution can be assigned to asset $i$.*

**Proof 3** *Consider a non-dominated solution $\Psi$ where asset $i$ is not protected and visited by vehicle $p$, with $i \in \mathcal{U}_p^{\Phi}$. Because the triangle inequality is satisfied, we can remove asset $i$ from the route of vehicle $p$ in $\Psi$ without delaying the visits of the other assets of the route. The solution built when removing asset $i$ from the route of vehicle $p$ in $\Psi$ is a feasible solution with the same total protected value and a strictly lower deviation, thus dominating the initial solution $\Psi$.* ■

Per Proposition 3, a vehicle will not visit an unprotected asset if the vehicle was not assigned to this asset in the pre-disruption solution. Only unprotected assets can be visited outside of their time windows. Hence, out-of-window arcs are sufficient for visits of assets outside of their time window.

We designate $\mathcal{E}_p^v$ as the set of valid arcs for vehicle $p$, and $\mathcal{E}_p^o$ as the set of out-of-window arcs for vehicle $p$. The set $\mathcal{E}_p$ of arcs available for vehicle $p$ is defined as $\mathcal{E}_p = \mathcal{E}_p^v \bigcup \mathcal{E}_p^o$. Sets $\delta_p^+(i)$, and $\delta_p^-(i)$ are described based on the available arcs $\mathcal{E}_p$.

Table 4.3 lists the decision variables and their type. The $X$ variables no longer represent the travels of a type of vehicle but of a single vehicle, hence becoming binary. New

Table 4.3: Decision variables used in the mathematical formulation of D-APP

| Symbol | Definition |
|---|---|
| $X_{ijp}$ | binary, 1 if vehicle $p$ is traveling from location $i$ to location $j$, 0 otherwise |
| $Y_i$ | binary, 1 if asset $i$ is protected, 0 otherwise |
| $S_i$ | continuous, the start time of service at asset $i$ |
| $Z_{ip}^+$ | binary, 1 if asset $i$ is added to the assignment of vehicle $p$, 0 otherwise |
| $Z_{ip}^-$ | binary, 1 if asset $i$ is removed from the assignment of vehicle $p$, 0 otherwise |

$Z^+$ and $Z^-$ variables are introduced to represent the deviation. The mathematical model, hereafter denoted by (D-APP), is written as follows:

$$Maximize\ f_1 = \sum_{i \in V^a} v_i Y_i \tag{4.1}$$

$$Minimize\ f_2 = \sum_{p \in \mathcal{P}} \sum_{i \in V^a} Z^+_{ip} + Z^-_{ip} \tag{4.2}$$

$$\sum_{j \in \delta^+_p(i)} X_{ijp} = start_{ip} \quad \forall i \in V^d,\ p \in \mathcal{P} \tag{4.3}$$

$$\sum_{p \in \mathcal{P}} \sum_{i \in \delta^-_p(n)} X_{inp} = |\mathcal{P}| \tag{4.4}$$

$$\sum_{i \in \delta^-_p(k)} X_{ikp} = \sum_{j \in \delta^+_p(k)} X_{kjp} \quad \forall k \in V^a,\ p \in \mathcal{P} \tag{4.5}$$

$$\sum_{p \in \mathcal{P}} \sum_{i \in \delta^-_p(k)} X_{ikp} \mathbf{cap}_p \geqslant Y_k \mathbf{r}_k \quad \forall k \in V^a \tag{4.6}$$

$$S_i + t_{ijp} + a_i \leqslant S_j + M_1(1 - X_{ijp}) \quad \forall (i,j) \in \mathcal{E}_p,\ p \in \mathcal{P} \tag{4.7}$$

$$S_i \geqslant o_i - M_2(1 - Y_i) \quad \forall i \in V^a \tag{4.8}$$

$$S_i \leqslant c_i + M_3(1 - Y_i) \quad \forall i \in V^a \tag{4.9}$$

$$\sum_{i \in \delta^-_p(k)} X_{ikp} - \sum_{i \in V^-} x_{ikp} = Z^+_{kp} - Z^-_{kp} \quad \forall k \in V^a,\ p \in \mathcal{P} \tag{4.10}$$

$$Y_i \in \{0,1\} \quad \forall i \in V^a \tag{4.11}$$

$$X_{ijp} \in \{0,1\} \quad \forall p \in \mathcal{P},\ (i,j) \in \mathcal{E}_p \tag{4.12}$$

$$S_i \in \mathbb{R} \quad \forall i \in V \tag{4.13}$$

$$Z^+_{ip} \in \{0,1\},\ Z^-_{ip} \in \{0,1\} \quad \forall i \in V^a,\ p \in \mathcal{P} \tag{4.14}$$

Equations (4.1) and (4.2) represent our objectives. Function $f_1$ is the total protected value and function $f_2$ is the deviation from the pre-disruption assignments.

Constraints (4.3)-(4.5) are the flow constraints for each vehicle, from their initial depot to the sink node. Constraints (4.6) ensure that an asset can be protected only if the vehicles assigned to the asset collectively meet the requirements on each resource. Constraints (4.7) ensure time integrity when visiting two assets consecutively. If asset $j$ is directly visited after asset $i$ by a vehicle $p$, service at asset $j$ should not start before service at asset $i$ is over and vehicle $p$ has had enough time to travel from asset $i$ to asset $j$.

Constraints (4.8) and (4.9) ensure that the start time of service of any protected asset is within its time window.

Constraints (4.10) ensure that the decision variables associated with deviation $Z^+_{ip}$ and $Z^-_{ip}$ are correctly set to 1 when asset $i$ has been added to (respectively, removed from) the initial route of vehicle $p$.

Constraints (4.11)-(4.14) define the domain of the decision variables.

A solution of (D-APP) can be represented as a list of routes for each vehicle, and a start time of service for each asset. The route of vehicle $p$ is a list of assets visited in that order by the vehicle, starting at its depot (noted $depot_p$) and ending at the sink node (noted $sink$). Between its depot and the sink node, the vehicle visits $n_p$ assets. We will designate $\mathcal{I}^p$ as the ordered set of assets visited by vehicle $p$, and $i^p_1$, $i^p_2$, ..., $i^p_{n_p}$ its elements. The service of assets protected in the solution starts strictly within their time window. For vehicle $p$, we will represent its route as follows: ($depot_p$, $i^p_1$, $i^p_2$, ..., $i^p_{n_p}$, $sink$).

## 4.3 REFORMULATION

In this section, we introduce a dominance property on the routes of the vehicles based on the protection status of the assets. Then, based on this dominance property, we introduce a new mathematical formulation for our problem.

### 4.3.1 *Dominance property*

As presented in Section 4.2, the set $\mathcal{E}_p$ of arcs available for vehicle $p$ can be broken down into two sets: the valid arcs, and the out-of-window arcs. Out-of-window arcs ensure that unprotected assets can be visited outside of their time window if necessary. We observed that the impact of out-of-window arcs is important since they represent on average a third of the available arcs for a vehicle.

We propose a new mathematical formulation that will only consider valid arcs. This new formulation is based on the following dominance property on the solutions of (D-APP).

**Proposition 4** *For every non-dominated point on the Pareto front, at least one solution exists such that unprotected assets are visited after protected assets.*

**Proof 4** *Consider a non-dominated solution of (D-APP). We will build an equivalent solution of (D-APP) that follows the dominance property by considering the route of each vehicle individually. For any vehicle $p$, the route of the vehicle can be represented by the ordered set of assets $\mathcal{I}^p$.*

*For the protected assets in $\mathcal{I}^p$, we can keep their start time of service unchanged. If we keep the visits of the protected assets in the same order, only removing unprotected assets in-between, the necessary valid arcs are available because the triangle inequality is satisfied.*

*For the unprotected assets in $\mathcal{I}^p$, we may need to alter their start time of service. As these assets are not protected, the time of service is not restricted by constraints (4.8-4.9). Per Proposition 3, unprotected assets in $\mathcal{I}^p$ were visited by vehicle $p$ in the pre-disruption solution: hence, we have the necessary out-of-window arcs available to enter unprotected assets. By keeping the order on unprotected assets, it is straightforward to compute a new start time of service for these assets that satisfies the synchronization constraint between vehicles.*

*Therefore, we can build a valid route for each vehicle* $p$ *that first visits the protected assets of* $\mathcal{J}^p$ *and then the unprotected assets, with an updated start time of service for the unprotected assets.*

*In the solution that uses the new routes we built, the vehicles visit the same assets as in the initial non-dominated solution, and protected assets are visited in the same global order and with the same start time of service (thus, remain protected). Hence, the new solution has the same total protected value and deviation as the initial non-dominated solution, and satisfies the dominance property.* ■

In a non-dominated solution that satisfies the dominance property, the route of a vehicle can be divided into two parts: first the valid route, that uses only valid arcs to visit all the protected assets assigned to the vehicle, and then the ancillary route, that uses only out-of-window arcs to visit all the unprotected assets assigned to the vehicle.

The only constraint on the ancillary routes is the synchronization between vehicles assigned to the same asset since time windows are no longer enforced for unprotected assets and there is no limit on how late an asset can be visited. Due to the synchronization constraint, if two assets are visited by different vehicles, the assets are visited in the same order by the vehicles. We can build a global order on the unprotected assets that is followed by all the ancillary routes. Due to Proposition 3, this order on unprotected assets alone is sufficient to describe the ancillary routes of all vehicles. We can build the ancillary routes of all vehicles based on an order on the unprotected assets in polynomial time.

If we know which assets are not protected, any order on the unprotected assets leads to feasible ancillary routes. By construction, there is always an out-of-window arc between two unprotected assets assigned to the same vehicle, and the order ensures that the synchronization constraint is verified. As ancillary routes do not contribute to the total protected value nor to the deviation, all feasible ancillary routes are equivalent to each other.

Hence, we can focus on finding only the valid routes for the vehicles. We no longer need to route unprotected assets, thus eliminating the need to consider visits of assets outside of their time window. All the planned visits contribute towards the total protected value, more efficiently using time and resources within the allowed deviation. Additionally, we only plan visits at times where the crews would be safe: we leave to the decision-maker the handling of unprotected assets.

### 4.3.2 *Mathematical formulation*

We therefore propose a new mathematical formulation that aims at constructing the valid routes for every vehicle. Once we know the valid routes, we can determine the assets that are not protected in the solution and build the ancillary routes for all vehicles based on an order on these assets. From Proposition 3, the only assets that can be visited in the ancillary route of a vehicle are the assets assigned to vehicle $p$ in the pre-disruption solution. These assets do not contribute to the deviation if they are not protected and not in the valid route of the corresponding vehicle.

The set $\mathcal{E}_p$ of arcs available for vehicle $p$ is restricted to the set of valid arcs, $\mathcal{E}_p^v$. The sets $\delta_p^+(i)$ and $\delta_p^-(i)$ are updated accordingly.

The new model, hereafter denoted by (D-APP-V), is written as follows:

$$Maximize\ f_1 = \sum_{i \in V^a} v_i Y_i \tag{4.15}$$

$$Minimize\ f_2 = \sum_{p \in \mathcal{P}} \sum_{i \in V^a} Z_{ip}^+ + Z_{ip}^- \tag{4.16}$$

$$\sum_{j \in \delta_p^+(i)} X_{ijp} = start_{ip} \quad \forall i \in V^d,\ p \in \mathcal{P} \tag{4.17}$$

$$\sum_{p \in \mathcal{P}} \sum_{i \in \delta_p^-(n)} X_{inp} = |\mathcal{P}| \tag{4.18}$$

$$\sum_{i \in \delta_p^-(k)} X_{ikp} = \sum_{j \in \delta_p^+(k)} X_{kjp} \quad \forall k \in V^a,\ p \in \mathcal{P} \tag{4.19}$$

$$\sum_{i \in \delta_p^-(k)} X_{ikp} \leqslant Y_k \quad \forall p \in \mathcal{P},\ k \in V^a \tag{4.20}$$

$$\sum_{p \in \mathcal{P}} \sum_{i \in \delta_p^-(k)} X_{ikp} \mathbf{cap}_p \geqslant Y_k \mathbf{r}_k \quad \forall k \in V^a \tag{4.21}$$

$$S_i + t_{ijp} + a_i \leqslant S_j + M_{ijp}(1 - X_{ijp}) \quad \forall (i,j) \in \mathcal{E}_p,\ p \in \mathcal{P} \tag{4.22}$$

$$o_i \leqslant S_i \leqslant c_i \quad \forall i \in V^a \tag{4.23}$$

$$\sum_{i \in \delta_p^-(k)} X_{ikp} = Z_{kp}^+ - Z_{kp}^- \quad \forall p \in \mathcal{P},\ k \in \mathcal{U}_p^\Phi \tag{4.24}$$

$$(1 - Y_k) + \sum_{i \in \delta_p^-(k)} X_{ikp} - 1 = Z_{kp}^+ - Z_{kp}^- \quad \forall p \in \mathcal{P},\ k \in \mathcal{V}_p^\Phi \tag{4.25}$$

$$Y_i \in \{0,1\} \quad \forall i \in V^a \tag{4.26}$$

$$X_{ijp} \in \{0,1\} \quad \forall p \in \mathcal{P},\ (i,j) \in \mathcal{E}_p \tag{4.27}$$

$$S_i \in \mathbb{R} \quad \forall i \in V \tag{4.28}$$

$$Z_{ip}^+ \in \{0,1\},\ Z_{ip}^- \in \{0,1\} \quad \forall i \in V^a,\ p \in \mathcal{P} \tag{4.29}$$

Equations (4.15) and (4.16) representing the objective functions $f_1$ and $f_2$ remain unchanged from the initial formulation (D-APP).

Flow constraints (4.17)-(4.19) are equivalent to the flow constraints in (D-APP), but the sets $\mathcal{E}_p$ are now considering only valid arcs. Constraints (4.20) ensure that all the assets that are visited are protected and that only protected assets are visited, as we only consider the valid part of the routes. Constraints (4.21) remain unchanged from

(D-APP). Constraints (4.22) are updated with big-M values specific to the arc, $M_{ijp}$. We define $M_{ijp} = maximum(o_i - c_j + a_i + t_{ijp}, 0)$.

Time window constraints are replaced by constraints (4.23) that enforce that the start time of service of an asset is always within its time window.

Deviation measurement constraints are split into two sets of constraints. Constraints (4.24), for assets not visited by the vehicle in the pre-disruption solution, follow the same pattern as constraints (4.10) in the initial formulation. Constraints (4.25), for assets visited by the vehicle in the pre-disruption solution, are updated to ensure that assets that can be visited in the ancillary route of the vehicle do not contribute to the deviation. In these constraints, we replaced the expression $\sum x_{ikp}$ by its fixed value (0 in (4.24), 1 in (4.25)). Constraints (4.26)-(4.29) define the domain of the decision variables.

## 4.4 VALID INEQUALITIES

In this section, we extend the valid inequalities presented for APP in Section 3.3 to the D-APP.

### 4.4.1 *Link between protection and deviation*

In a non-dominated solution $\Psi$, we can link the protection status of an asset to changes in the vehicles assigned to the asset. If an asset $i$ has been added to the initial route of at least one vehicle in the solution $\Psi$, there are no longer only vehicles already assigned to the asset in the pre-disruption solution $\Phi$ that are assigned to the asset. On the basis of Proposition 3, asset $i$ then must be protected in $\Psi$. We introduce the following corollary of Proposition 3:

**Corollary 1** *In a non-dominated solution, if asset $i$ has been added to the initial route of at least one vehicle, then asset $i$ is protected in the solution.*

Hence, we know that if a variable $Z^+_{ip} = 1$, then $Y_i = 1$. Proposition 5 shows a similar link for $Z^-$ variables, i.e., when a vehicle has been removed from the initial route of a vehicle.

**Proposition 5** *In a non-dominated solution, if asset $i$ has been removed from the initial route of at least one vehicle, then asset $i$ is protected in the solution.*

**Proof 5** *Consider a non-dominated solution $\Psi$ where asset $i$ is not protected and removed from the initial route of vehicle $p$. Because unprotected assets can be visited outside of their time windows, we can add asset $i$ back to the route of vehicle $p$ in $\Psi$ without delaying the visits of the other assets of the route. The solution built when adding asset $i$ to the route of vehicle $p$ in $\Psi$ is a feasible solution with the same total protected value and a strictly lower deviation, thus dominating the initial solution $\Psi$.* ■

We note that an asset cannot be simultaneously added and removed from the initial route of the same vehicle. On the basis of Corollary 1 and Proposition 5, the following set of inequalities is valid:

$$Y_i \geqslant Z^+_{ip} + Z^-_{ip} \quad \forall p \in \mathcal{P} \tag{4.30}$$

### 4.4.2 *Deviation-based inequalities*

We want to extend the valid inequalities based on bounds presented in Section 3.3.1. Proposition 1 does not hold for D-APP: as assets can be visited outside of their time window in order to avoid deviation, there can be a non-dominated solution where an asset is visited but not protected. It is not possible to use the valid inequalities for APP in their current form. However, we can note that Corollary 1 and Proposition 5 are close in substance to Proposition 1. We no longer seek bounds on the number of vehicles assigned to an asset but rather on the number of vehicles added to or removed from protecting an asset. We can note that, contrary to the valid inequalities on APP, the valid inequalities are based on a property of a non-dominated solution and not on a dominance relation.

*Lower bound on vehicle additions for protection*

If asset $i$ is protected, there are at least $lb_v(i)$ vehicles entering the asset or, equivalently, assigned to the asset. A vehicle assigned to the asset contributes to the deviation only if the asset was not in the initial route of the vehicle. The following set of inequalities is valid:

$$\sum_{p \in \mathcal{P}} Z^+_{ip} \geqslant (lb_v(i) - |\mathcal{P}^\Phi_i|) Y_i \quad \forall i \in V^a \tag{4.31}$$

*Upper bound on vehicle additions for protection*

**Proposition 6** *In a non-dominated solution, if an asset is added to the initial route of a vehicle, the vehicle is not redundant.*

**Proof 6** *Consider a non-dominated solution $\Psi$ where asset $i$ was added to the initial route of vehicle $p$ and vehicle $p$ is redundant to the protection of asset $i$. Because vehicle $p$ is redundant, the resource requirement of asset $i$ is still met if we remove asset $i$ from the route of vehicle $p$ in $\Psi$. Removing an asset from the route of a vehicle does not impact the time of visit for the remaining assets on the route, due to the triangle inequality. The solution built when removing asset $i$ from the route of vehicle $p$ in $\Psi$ is a feasible solution with the same total protected value and strictly lower deviation, thus dominating the initial solution $\Psi$.* ■

However, a vehicle already assigned to asset $i$ in the pre-disruption solution $\Phi$ can be redundant. Removing asset $i$ from the initial route of such a vehicle would increase the deviation without changing the total protected value. Let $ub^+_v(i)$ be an upper bound on the number of vehicles for which asset $i$ can be added to the initial route without any of these vehicles being redundant. The following set of valid inequalities thus holds:

$$\sum_{p \in \mathcal{P}} Z^+_{ip} \leqslant ub^+_v(i) Y_i \quad \forall i \in V^a \tag{4.32}$$

COMPUTATION OF $ub^+_v(i)$ We introduce a MIP to compute $ub^+_v(i)$ for a given asset $i$, a set of vehicles $\mathcal{P}$, and a subset of $\mathcal{P}$ designated as $\Lambda$. For our purposes, we would use $\Lambda = \overline{\mathcal{P}^{\Phi}_i}$. We note $cap_{pu}$ the $u$-th resource of capability vector $\mathbf{cap}_p$ and $r_{iu}$ the $u$-th resource of requirement vector $r_i$. For each vehicle $p \in \mathcal{P}$, a binary decision variable $W_p$ is set to 1 if vehicle $p$ is assigned to the protection of the asset. For each vehicle $p \in \mathcal{P}$ and each resource $u$, a binary decision variable $Y_{pu}$ is equal to 1 if resource $u$ is still covered if we remove vehicle $p$. We aim at finding the maximum number of vehicles in $\Lambda$ needed to cover the resource requirement of the asset without any of the vehicles being redundant.

$$Maximize \sum_{p \in \Lambda} W_p \tag{4.33}$$

$$\sum_{p \in \mathcal{P}} W_p \mathbf{cap}_p \geqslant \mathbf{r}_i \tag{4.34}$$

$$\sum_{p' \in \mathcal{P} \setminus \{p\}} W_{p'} cap_{p'u} \leqslant r_{iu} - 1 + M_{pu} Y_{pu} \quad \forall u = 1, ..., K,\ p \in \mathcal{P} \tag{4.35}$$

$$\sum_{u=1}^{K} Y_{pu} \leqslant K - W_p \quad \forall p \in \mathcal{P} \tag{4.36}$$

$$W_p \in \{0, 1\} \quad \forall p \in \mathcal{P} \tag{4.37}$$

$$Y_{pu} \in \{0, 1\} \quad \forall u = 1, ..., K,\ p \in \mathcal{P} \tag{4.38}$$

Objective (4.33) maximizes the number of vehicles in $\Lambda$ assigned to the asset. Constraint (4.34) ensures that the cumulative capability vector of the selected vehicles covers the resource requirement of the asset.

Constraints (4.35) and Constraints (4.36) ensure that if a vehicle $p \in \mathcal{P}$ is selected, at least one resource is no longer covered when vehicle $p$ is removed. In other words, if a vehicle is selected, it is not redundant. To ensure that constraints (4.35) are verified when the $u$-th resource is still covered or when vehicle $p$ is not selected, we define:

$$M_{pu} = 1 - r_{pu} + \sum_{p' \in \mathcal{P} \setminus \{p\}} cap_{p'u}$$

Constraints (4.37) and (4.38) define the domain of the decision variables.

We can note that MIP (3.18-3.23) used to compute $ub_v(i)$, the upper bound on the number of vehicles required for protecting asset $i$, is a special case of this MIP with $\Lambda = \mathcal{P}$.

*Bounds on vehicle removals for protection*

The following set of valid inequalities provides bounds on the deviation due to removals from initial routes implied by the protection of the asset. We first note that there is no interesting lower bound in that case. As we already noted, in a non-dominated solution, vehicles assigned to an asset $i$ can be redundant.

Vehicles not assigned to asset $i$ in the pre-disruption solution $\Phi$ may not be sufficient to meet the resource requirement of the asset. In this case, asset $i$ can only be protected also using some of the vehicles already assigned to the asset in $\Phi$. Let $\alpha_v^{\Phi}(i)$ be a lower bound on the number of vehicles assigned to asset $i$ in $\Phi$ that need to remain assigned to asset $i$ in order to protect it. The following set of inequality is valid:

$$\sum_{p \in \mathcal{P}} Z_{ip}^{-} \leqslant (|\mathcal{P}_i^{\Phi}| - \alpha_v^{\Phi}(i)) Y_i \quad \forall i = m+1, ..., n-1 \tag{4.39}$$

COMPUTATION OF $\alpha_v^{\Phi}(i)$ We introduce a MIP to compute $\alpha_v^{\Phi}(i)$ for a given asset $i$, a set of available vehicles $\mathcal{P}$ and a partition of $\mathcal{P}$ in two subsets $\mathcal{P}_i^{\Phi}$ and $\overline{\mathcal{P}_i^{\Phi}}$. We suppose that the cumulative capability vectors of vehicles in $\overline{\mathcal{P}_i^{\Phi}}$ does not meet the resource requirement of asset $i$. Otherwise, if the cumulative capability vector meets the resource requirement of asset $i$, we can obtain a straightforward bound. We can carry out this test by summing the capability vectors of these vehicles, in $O(|\overline{\mathcal{P}_i^{\Phi}}|.K)$, where $K$ is the number of resources. In that case, it is possible to remove the asset from the initial routes of all vehicles in $\mathcal{P}_i^{\Phi}$ and still meet the resource requirement. For these assets, we have $\alpha_v^{\Phi}(i) = 0$.

For each vehicle $p \in \mathcal{P}$, a binary variable decision $W_p$ is set to 1 if vehicle $p$ is assigned to the protection of asset $i$. We aim at minimizing the number of vehicles in $\mathcal{P}_i^{\Phi}$ assigned to the protection of the asset.

$$\text{Minimize} \sum_{p \in \mathcal{P}_i^{\Phi}} W_p \tag{4.40}$$

$$\sum_{p \in \mathcal{P}} W_p \mathbf{cap}_p \geqslant \mathbf{r}_i \tag{4.41}$$

$$W_p \in \{0, 1\} \quad \forall p \in \mathcal{P} \tag{4.42}$$

Objective (4.40) minimizes the number of vehicles in $\mathcal{P}_i^{\Phi}$ assigned to the asset. Hence, a priority is given to the vehicles that are in $\overline{\mathcal{P}_i^{\Phi}}$. Constraint (4.41) ensures that the cumulative capability vector of the selected vehicles meets the resource requirement of the asset. Constraints (4.42) define the domain of the decision variables.

### 4.4.3 *Incompatibility cliques*

We can extend the notion of incompatibility for a vehicle type between two assets to incompatibility for a single vehicle. Let $G_p^{inc/v} = (V^a, E_p^{inc/v})$ be the graph of incompatibilities for vehicle $p$ between assets, where :

$$E_p^{inc/v} = \{(i,j) \mid i,j \in V^a \ : \ o_i + a_i + t_{ijp} > c_j \ \wedge \ \ o_j + a_j + t_{jip} > c_i\}$$

Consider a clique $\mathcal{C}$ of graph $G_p^{inc/v}$. At most one asset not visited by vehicle $p$ in the pre-disruption solution $\Phi$ can be visited by vehicle $p$. Otherwise, according to Corollary 1, vehicle $p$ would visit two different protected assets of the clique, which is impossible. The following inequality is thus valid for any clique $\mathcal{C}$:

$$\sum_{j \in \mathcal{C} \cap \mathcal{U}_p^\Phi} \sum_{i \in \delta_p^-(j)} X_{ijp} \leqslant 1 \tag{4.43}$$

Vehicle $p$ can however visit multiple assets of the clique if only at most one is protected. As visits of unprotected assets can happen outside of their time window, incompatibility/vehicle for unprotected asset can be disregarded. Per Proposition 3, a vehicle can be assigned to an unprotected asset only if the vehicle was assigned to the asset in the pre-disruption solution. A vehicle can thus be assigned to every unprotected asset plus one protected asset within a clique. Hence, the following inequality is valid for any clique $\mathcal{C}$:

$$\sum_{j \in \mathcal{C}} \sum_{i \in \delta_p^-(j)} X_{ijp} \leqslant 1 + \sum_{j \in \mathcal{C} \cap \mathcal{V}_p^\Phi} (1 - Y_j) \tag{4.44}$$

The reformulation we presented changed the way visits outside of time windows are handled by the model. An asset visited outside of its time window is never visited using a valid arc. For a clique $\mathcal{C}$ of $G_p^{inc/v}$, we can generalize Equations (4.43) and (4.44) to all the assets of the clique. The following inequality is valid for (D-APP-V) only:

$$\sum_{j \in \mathcal{C}} \sum_{i \in \delta_p^-(j)} X_{ijp} \leqslant 1 \tag{4.45}$$

*Strict incompatibility*

Strict incompatibility between assets can be defined using incompatibilities for all vehicles. Clique protection cuts (Equation (3.25)) are also valid for (D-APP) and (D-APP-V) in their current form.

## 4.5 COMPUTATIONAL EXPERIMENTS

We carried out computational testing on a computer with an Intel Core i7-8550U processor and 8 GB of RAM. We implemented the models in Julia, and solved the problems using CPLEX 12.10.

We will consider 10 benchmark instances[1]. We generated 10 custom benchmark instances with 100 assets following the instructions provided by Merwe et al. (2015). The assets are randomly distributed on a 80km by 80km grid. Custom instances with less than 100 assets are created using a subset of the original instances. Time windows have a fixed length for every asset. The vehicles are initially available in a central depot. The capability vectors of vehicles for our custom instances are given in Table 3.1. For each instance, we calculated the pre-disruption vehicle assignment with all ten vehicles available.

In the following, we consider as disruption the breakdown of a vehicle before the start of operations. Hence, every vehicle starts at the central depot. We note that we could consider a disruption that occurs after the start of operations by using the location of the vehicles at the time of the disruption as their initial depots.

### 4.5.1 *Solution method*

We used the $\epsilon$-constraint method introduced by Ozlen et al. (2009) to generate the Pareto front, i.e., the entire trade-off surface. In this method, we rewrite the objective function as $Maximize\ f_1 - w_2 f_2$ where $w_2$ is such that the optimal solution is the solution with highest possible protected value $f_1$ and offers a lexicographical ordering on deviation $f_2$. In other words, solving with this objective function always gives a non-dominated point of the Pareto front. We use $w_2 = \frac{1}{f_2^{GUB} - f_2^{GLB} + 1}$, where $f_2^{GUB}$ and $f_2^{GLB}$ are the upper bound and lower bound on the deviation for any feasible solution. In order to generate all the points of the Pareto front, we introduce a new constraint $f_2 \leqslant \epsilon$ that limits the value of the deviation to $\epsilon$. The initial value of $\epsilon$ is an upper bound on the highest possible deviation. At each step, we replace the value of $\epsilon$ by the value of the deviation $f_2$ found at the last iteration minus one. When $\epsilon$ is lower than zero (the lowest possible deviation), the algorithms ends and returns all of the non-dominated points of the Pareto front. We provide a description of the method in Algorithm 1.

The first step of the $\epsilon$-constraint method is the computation of the extreme point with maximum deviation.

BOUNDS COMPUTATION The global upper bound on deviation $f_2^{GUB}$ is set to the sums of the upper bounds on variables $Z^+$ and $Z^-$ in inequalities (4.32) and (4.39).

$$f_2^{GUB} = \sum_{i \in V^a} \left( ub_v^+(i) + (|\mathcal{P}_i^\Phi| - \alpha_v^\Phi(i)) \right) \tag{4.46}$$

[1] See https://www.hds.utc.fr/~penaquen/dokuwiki/doku.php for the detailed instances, initial routes and best known results.

**Algorithm 1:** $\epsilon$-constraint method

1: $ND = \emptyset$
2: Compute bounds on deviation $f_2^{GLB}$ and $f_2^{GUB}$
3: Let $w_2 = \frac{1}{f_2^{GUB} - f_2^{GLB} + 1}$
4: Let $\epsilon = f_2^{GUB}$
5: **while** $\epsilon \geqslant 0$ **do**
6: Solve the model with objective $max\ f_1 - w_2 f_2$ and constraint $f_2 \leqslant \epsilon$
7: Let $x^*$ be the optimal solution
8: $ND = ND \cup \{x^*\}$
9: $\epsilon = f_2(x^*) - 1$
10: **end while**
11: **return** ND

The global lower bound on deviation $f_2^{GLB}$ is set to 0.

### 4.5.2 *Extreme point generation*

In order to compare our new formulation with the results obtained by Merwe et al. (2017), we focus on finding an optimal solution with maximum deviation allowed, i.e., the extreme point that maximizes total protected value. We computed the extreme point for each of the ten custom instances, with three different random vehicle breakdowns, for sizes from 30 to 60 assets, using each of the MIP formulations we presented. We set a time limit of 1800 seconds (30 minutes). We provided a warm-start solution to CPLEX by setting the values of X variables to their value x in the initial solution $\Phi$.

Tables 4.4, 4.5, 4.6 and 4.7 give the detailed results obtained when generating the extreme point for the custom benchmark instances, with 30, 40, 50 and 60 assets, respectively. For each instance, column "br" shows the vehicle chosen for breakdown. Column "ST" shows the solution time in seconds, the objective values for the extreme point are given in column "TPV" for the total protected value, given as a percentage of the total asset value, and "Dev" for the deviation. When the time limit is reached, the objective values for the best feasible solution found is given. The "Gap" column shows the relative optimality gap.

We can see from the tables that the reformulation outperforms the initial formulation for all instances. In particular, we can see in Tables 4.6 and 4.7 that the time limit is reached less often with the reformulation.

Table 4.8 presents the aggregated results obtained by the initial formulation and reformulation, with and without the valid inequalities, for the custom instances. Column "ST" shows the average solution time in seconds. A dash indicates that no feasible solution was found for any of the instances within the time limit. Column "TPV" shows the average total protected value of the solution returned by CPLEX, as a percentage of the total asset value. The average relative optimality gap is shown in brackets.

Table 4.4: Detailed results for extreme point generation for custom instances with 30 assets

| Instance | br | (D-APP) | | | | (D-APP) + VI | | | | (D-APP-V) | | | | (D-APP-V) + VI | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap |
| m101 | 1 | 3.4 | 100.0 | 9 | 0 | **0.91** | **100.0** | **9** | **0** | 5.5 | 100.0 | 9 | 0 | 1.2 | 100.0 | 9 | 0 |
| m101 | 3 | **0.7** | **100.0** | **9** | **0** | 0.84 | 100.0 | 9 | 0 | 0.78 | 100.0 | 15 | 0 | 1.2 | 100.0 | 15 | 0 |
| m101 | 9 | **0.52** | **100.0** | **3** | **0** | 0.73 | 100.0 | 3 | 0 | 0.54 | 100.0 | 6 | 0 | 1.1 | 100.0 | 6 | 0 |
| m102 | 1 | *1800* | *88.97* | *19* | *0.02* | 2.9 | 88.97 | 19 | 0 | **0.93** | **88.97** | **18** | **0** | 1.4 | 88.97 | 18 | 0 |
| m102 | 4 | *1800* | *84.36* | *9* | *8.49* | 11 | 84.36 | 9 | 0 | **0.59** | **84.36** | **11** | **0** | 1.1 | 84.36 | 11 | 0 |
| m102 | 8 | 261 | 88.97 | 7 | 0 | 1.4 | 88.97 | 7 | 0 | **0.94** | **88.97** | **12** | **0** | 1.4 | 88.97 | 12 | 0 |
| m103 | 5 | 4.4 | 100.0 | 10 | 0 | 1.6 | 100.0 | 10 | 0 | **0.86** | **100.0** | **9** | **0** | 1.1 | 100.0 | 9 | 0 |
| m103 | 6 | **0.72** | **100.0** | **7** | **0** | 0.98 | 100.0 | 7 | 0 | 0.76 | 100.0 | 6 | 0 | 1.0 | 100.0 | 6 | 0 |
| m103 | 10 | 1.0 | 100.0 | 4 | 0 | 0.94 | 100.0 | 4 | 0 | **0.64** | **100.0** | **1** | **0** | 0.91 | 100.0 | 1 | 0 |
| m104 | 1 | 2.8 | 100.0 | 10 | 0 | 2.9 | 100.0 | 10 | 0 | **1.2** | **100.0** | **9** | **0** | 1.7 | 100.0 | 9 | 0 |
| m104 | 5 | 1.2 | 100.0 | 8 | 0 | **1.2** | **100.0** | **8** | **0** | 1.4 | 100.0 | 13 | 0 | 1.7 | 100.0 | 13 | 0 |
| m104 | 9 | 3.3 | 100.0 | 7 | 0 | 2.2 | 100.0 | 7 | 0 | **1.1** | **100.0** | **9** | **0** | 1.7 | 100.0 | 9 | 0 |
| m105 | 6 | 1.4 | 95.9 | 5 | 0 | 1.1 | 95.9 | 5 | 0 | **0.59** | **95.9** | **7** | **0** | 1.1 | 95.9 | 7 | 0 |
| m105 | 8 | 1.1 | 95.9 | 8 | 0 | 1.0 | 95.9 | 8 | 0 | **0.89** | **95.9** | **3** | **0** | 2.8 | 95.9 | 3 | 0 |
| m105 | 10 | 1.7 | 95.9 | 6 | 0 | 1.5 | 95.9 | 6 | 0 | 1.7 | 95.9 | 4 | 0 | **1.0** | **95.9** | **4** | **0** |
| m106 | 3 | 868 | 98.21 | 10 | 0 | 5.2 | 98.21 | 10 | 0 | **1.6** | **98.21** | **12** | **0** | 1.8 | 98.21 | 12 | 0 |
| m106 | 5 | 1637 | 98.21 | 13 | 0 | 6.4 | 98.21 | 13 | 0 | **2.1** | **98.21** | **21** | **0** | 2.3 | 98.21 | 21 | 0 |
| m106 | 8 | *1800* | *98.21* | *1* | *1.29* | **1.2** | **98.21** | **1** | **0** | 2.9 | 98.21 | 6 | 0 | 1.2 | 98.21 | 6 | 0 |
| m107 | 1 | 12 | 95.9 | 8 | 0 | 1.2 | 95.9 | 8 | 0 | **0.6** | **95.9** | **10** | **0** | 1.4 | 95.9 | 10 | 0 |
| m107 | 3 | 47 | 95.9 | 12 | 0 | 1.2 | 95.9 | 12 | 0 | **0.6** | **95.9** | **10** | **0** | 1.2 | 95.9 | 10 | 0 |
| m107 | 10 | 7.1 | 95.9 | 7 | 0 | 1.1 | 95.9 | 7 | 0 | **0.59** | **95.9** | **13** | **0** | 1.2 | 95.9 | 13 | 0 |
| m108 | 2 | 10 | 99.23 | 8 | 0 | 1.2 | 99.23 | 8 | 0 | **0.88** | **99.23** | **16** | **0** | 1.4 | 99.23 | 16 | 0 |
| m108 | 4 | *1800* | *96.41* | *9* | *1.57* | 2.4 | 96.41 | 9 | 0 | **0.53** | **96.41** | **9** | **0** | 1.2 | 96.41 | 9 | 0 |
| m108 | 5 | *1801* | *96.41* | *9* | *1.58* | 3.5 | 96.41 | 8 | 0 | **0.49** | **96.41** | **5** | **0** | 0.98 | 96.41 | 5 | 0 |
| m109 | 1 | 14 | 97.69 | 10 | 0 | 2.4 | 97.69 | 10 | 0 | **0.81** | **97.69** | **15** | **0** | 1.7 | 97.69 | 15 | 0 |
| m109 | 4 | *1802* | *95.9* | *11* | *1.87* | 14 | 95.9 | 11 | 0 | **1.2** | **95.9** | **12** | **0** | 1.2 | 95.9 | 12 | 0 |
| m109 | 10 | 68 | 97.69 | 6 | 0 | 12 | 97.69 | 6 | 0 | **0.83** | **97.69** | **11** | **0** | 1.5 | 97.69 | 11 | 0 |
| m110 | 4 | 95 | 91.03 | 14 | 0 | 1.3 | 91.03 | 14 | 0 | **1.0** | **91.03** | **12** | **0** | 1.0 | 91.03 | 12 | 0 |
| m110 | 7 | 0.92 | 91.03 | 7 | 0 | 0.94 | 91.03 | 7 | 0 | **0.79** | **91.03** | **8** | **0** | 1.1 | 91.03 | 8 | 0 |
| m110 | 9 | 1.2 | 91.03 | 3 | 0 | 1.3 | 91.03 | 3 | 0 | **0.53** | **91.03** | **7** | **0** | 0.99 | 91.03 | 7 | 0 |

Table 4.5: Detailed results for extreme point generation for custom instances with 40 assets

| Instance | br | (D-APP) | | | (D-APP) + VI | | | | (D-APP-V) | | | (D-APP-V) + VI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap |
| m101 | 2 | 197 | 94.4 | 16 | 0 | 5.0 | 94.4 | 16 | 0 | **2.1** | **94.4** | **15** | **0** | 2.7 | 94.4 | 15 | 0 |
| m101 | 3 | 99 | 94.4 | 12 | 0 | 5.2 | 94.4 | 12 | 0 | **1.7** | **94.4** | **14** | **0** | 2.3 | 94.4 | 14 | 0 |
| m101 | 4 | 28 | 94.4 | 15 | 0 | 2.7 | 94.4 | 15 | 0 | **1.1** | **94.4** | **16** | **0** | 1.7 | 94.4 | 16 | 0 |
| m102 | 2 | *1808* | *81.59* | *13* | *17.48* | *1801* | *88.27* | *28* | *4.48* | **106** | **88.27** | **38** | **0** | 115 | 88.27 | 38 | 0 |
| m102 | 8 | *1801* | *89.71* | *8* | *6.84* | 724 | 90.79 | 16 | 0 | 77 | 90.79 | 17 | 0 | **14** | **90.79** | **17** | **0** |
| m102 | 10 | *1804* | *87.55* | *7* | *9.48* | 106 | 90.79 | 16 | 0 | **15** | **90.79** | **25** | **0** | 22 | 90.79 | 25 | 0 |
| m103 | 5 | *1800* | *98.56* | *18* | *0.91* | 318 | 98.56 | 18 | 0 | **1.5** | **98.56** | **16** | **0** | 1.8 | 98.56 | 16 | 0 |
| m103 | 6 | 49 | 99.46 | 15 | 0 | 29 | 99.46 | 15 | 0 | **0.94** | **99.46** | **15** | **0** | 1.8 | 99.46 | 15 | 0 |
| m103 | 9 | 124 | 99.46 | 12 | 0 | 30 | 99.46 | 12 | 0 | **1.0** | **99.46** | **12** | **0** | 1.8 | 99.46 | 12 | 0 |
| m104 | 4 | 28 | 100.0 | 29 | 0 | 7.7 | 100.0 | 29 | 0 | **2.1** | **100.0** | **22** | **0** | 2.8 | 100.0 | 22 | 0 |
| m104 | 5 | 25 | 100.0 | 32 | 0 | 23 | 100.0 | 32 | 0 | 3.6 | 100.0 | 25 | 0 | **3.1** | **100.0** | **25** | **0** |
| m104 | 8 | 21 | 100.0 | 19 | 0 | 26 | 100.0 | 19 | 0 | **1.0** | **100.0** | **7** | **0** | 2.0 | 100.0 | 7 | 0 |
| m105 | 1 | 29 | 97.11 | 15 | 0 | 3.8 | 97.11 | 15 | 0 | **0.93** | **97.11** | **11** | **0** | 1.6 | 97.11 | 11 | 0 |
| m105 | 7 | 21 | 97.11 | 13 | 0 | 3.8 | 97.11 | 13 | 0 | **0.92** | **97.11** | **6** | **0** | 1.8 | 97.11 | 6 | 0 |
| m105 | 10 | 81 | 97.11 | 11 | 0 | 4.0 | 97.11 | 11 | 0 | **0.76** | **97.11** | **5** | **0** | 1.5 | 97.11 | 5 | 0 |
| m106 | 4 | *1809* | *96.39* | *15* | *3.37* | *1801* | *96.57* | *16* | *1.85* | 9.5 | 96.57 | 19 | 0 | **8.4** | **96.57** | **19** | **0** |
| m106 | 7 | *1828* | *97.83* | *10* | *1.85* | 1493 | 97.83 | 10 | 0 | 9.9 | 97.83 | 10 | 0 | **6.7** | **97.83** | **10** | **0** |
| m106 | 10 | *1801* | *96.93* | *9* | *2.79* | *1801* | *97.11* | *10* | *0.35* | **3.0** | **97.11** | **16** | **0** | 6.8 | 97.11 | 16 | 0 |
| m107 | 3 | *1801* | *97.11* | *20* | *0.01* | 19 | 97.11 | 20 | 0 | 3.7 | 97.11 | 18 | 0 | **3.4** | **97.11** | **18** | **0** |
| m107 | 8 | 23 | 97.11 | 10 | 0 | 3.2 | 97.11 | 10 | 0 | **3.0** | **97.11** | **19** | **0** | 3.1 | 97.11 | 19 | 0 |
| m107 | 10 | 126 | 97.11 | 10 | 0 | **3.7** | **97.11** | **10** | **0** | 10 | 97.11 | 28 | 0 | 9.0 | 97.11 | 28 | 0 |
| m108 | 2 | *1800* | *98.01* | *16* | *1.47* | 3.6 | 98.01 | 16 | 0 | **1.3** | **98.01** | **15** | **0** | 1.8 | 98.01 | 15 | 0 |
| m108 | 6 | *1800* | *98.01* | *10* | *1.47* | 5.7 | 98.01 | 10 | 0 | **0.87** | **98.01** | **10** | **0** | 1.9 | 98.01 | 10 | 0 |
| m108 | 7 | *1800* | *98.01* | *7* | *1.46* | 3.2 | 98.01 | 7 | 0 | **0.84** | **98.01** | **3** | **0** | 1.6 | 98.01 | 3 | 0 |
| m109 | 6 | 396 | 98.38 | 18 | 0 | 6.5 | 98.38 | 18 | 0 | **1.8** | **98.38** | **12** | **0** | 2.3 | 98.38 | 12 | 0 |
| m109 | 7 | 168 | 98.38 | 17 | 0 | 6.3 | 98.38 | 17 | 0 | **1.2** | **98.38** | **8** | **0** | 2.0 | 98.38 | 8 | 0 |
| m109 | 9 | 19 | 98.38 | 14 | 0 | 3.3 | 98.38 | 14 | 0 | **1.6** | **98.38** | **12** | **0** | 2.1 | 98.38 | 12 | 0 |
| m110 | 5 | *1800* | *90.79* | *4* | *2.78* | 2.3 | 90.79 | 4 | 0 | **1.4** | **90.79** | **8** | **0** | 1.5 | 90.79 | 8 | 0 |
| m110 | 6 | 22 | 93.68 | 8 | 0 | 3.2 | 93.68 | 8 | 0 | 3.1 | 93.68 | 20 | 0 | **2.1** | **93.68** | **20** | **0** |
| m110 | 7 | 5.6 | 93.68 | 9 | 0 | **2.4** | **93.68** | **9** | **0** | 9.5 | 93.68 | 26 | 0 | 6.3 | 93.68 | 26 | 0 |

Table 4.6: Detailed results for extreme point generation for custom instances with 50 assets

| Instance | br | (D-APP) | | | | (D-APP) + VI | | | | (D-APP-V) | | | | (D-APP-V) + VI | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap |
| m101 | 3 | 607 | 95.48 | 16 | 0 | 12 | 95.48 | 16 | 0 | **4.9** | **95.48** | **27** | **0** | 7.1 | 95.48 | 27 | 0 |
| m101 | 4 | *1800* | *95.48* | *25* | *0.15* | 137 | 95.48 | 25 | 0 | **7.3** | **95.48** | **22** | **0** | 8.4 | 95.48 | 22 | 0 |
| m101 | 6 | *1801* | *95.48* | *16* | *0.15* | 15 | 95.48 | 15 | 0 | **2.0** | **95.48** | **12** | **0** | 2.4 | 95.48 | 12 | 0 |
| m102 | 6 | *1809* | *83.05* | *8* | *16.66* | *1800* | *87.01* | *17* | *0.32* | 930 | 87.01 | 16 | 0 | **150** | **87.01** | **16** | **0** |
| m102 | 8 | *1801* | *83.05* | *11* | *16.66* | *1800* | *86.86* | *19* | *1.77* | **769** | **87.01** | **11** | **0** | 775 | 87.01 | 11 | 0 |
| m102 | 10 | *1807* | *83.19* | *10* | *16.46* | *1801* | *84.75* | *11* | *4.31* | 191 | 87.01 | 23 | 0 | **40** | **87.01** | **23** | **0** |
| m103 | 2 | *1800* | *98.87* | *16* | *0.71* | 166 | 98.87 | 17 | 0 | **3.4** | **98.87** | **18** | **0** | 4.6 | 98.87 | 18 | 0 |
| m103 | 7 | *1801* | *98.87* | *12* | *0.71* | 261 | 98.87 | 13 | 0 | **2.0** | **98.87** | **10** | **0** | 2.9 | 98.87 | 10 | 0 |
| m103 | 9 | 684 | 99.58 | 21 | 0 | 97 | 99.58 | 21 | 0 | **3.0** | **99.58** | **16** | **0** | 4.8 | 99.58 | 16 | 0 |
| m104 | 3 | *1800* | *97.6* | *21* | *0.15* | 154 | 97.74 | 38 | 0 | 88 | 97.74 | 53 | 0 | **81** | **97.74** | **53** | **0** |
| m104 | 4 | *1800* | *95.9* | *11* | *1.91* | *1802* | *96.61* | *12* | *0.93* | 25 | 96.61 | 19 | 0 | **14** | **96.61** | **19** | **0** |
| m104 | 5 | *1801* | *92.23* | *23* | *5.97* | *1801* | *96.61* | *25* | *1.02* | 112 | 96.61 | 18 | 0 | **24** | **96.61** | **18** | **0** |
| m105 | 3 | 450 | 97.74 | 28 | 0 | 11 | 97.74 | 28 | 0 | **2.2** | **97.74** | **20** | **0** | 3.1 | 97.74 | 20 | 0 |
| m105 | 5 | 698 | 97.74 | 21 | 0 | 7.3 | 97.74 | 21 | 0 | **2.8** | **97.74** | **25** | **0** | 3.6 | 97.74 | 25 | 0 |
| m105 | 10 | 258 | 97.74 | 15 | 0 | 8.2 | 97.74 | 15 | 0 | **1.9** | **97.74** | **10** | **0** | 2.7 | 97.74 | 10 | 0 |
| m106 | 1 | *1803* | *92.66* | 22 | *7.62* | *1803* | *95.34* | 27 | *1.47* | 12 | 95.48 | 24 | 0 | **10** | **95.48** | **24** | **0** |
| m106 | 4 | *1831* | *92.51* | *12* | *7.78* | *1800* | *94.35* | *19* | *2.24* | 17 | 95.06 | 28 | 0 | **16** | **95.06** | **28** | **0** |
| m106 | 7 | *1800* | *98.31* | 22 | *1.44* | *1807* | *97.74* | *20* | *1.01* | 15 | 98.31 | 26 | 0 | **14** | **98.31** | **26** | **0** |
| m107 | 1 | *1821* | *91.1* | *13* | *7.13* | 1638 | 96.89 | 32 | 0 | 51 | 96.89 | 36 | 0 | **47** | **96.89** | **36** | **0** |
| m107 | 7 | *1800* | *94.63* | *15* | *3.14* | 1211 | 96.89 | 33 | 0 | 10 | 96.89 | 11 | 0 | **6.3** | **96.89** | **11** | **0** |
| m107 | 9 | *1801* | *94.49* | *13* | *3.29* | *1801* | *96.89* | 23 | *0.28* | **18** | **96.89** | **13** | **0** | 33 | 96.89 | 13 | 0 |
| m108 | 1 | *1801* | *98.45* | 24 | *1.15* | 13 | 98.45 | 24 | 0 | **2.9** | **98.45** | **19** | **0** | 3.7 | 98.45 | 19 | 0 |
| m108 | 7 | *1806* | *98.45* | *8* | *1.14* | 6.9 | 98.45 | 8 | 0 | **1.7** | **98.45** | **12** | **0** | 3.0 | 98.45 | 12 | 0 |
| m108 | 8 | *1801* | *98.45* | *8* | *1.14* | 4.5 | 98.45 | 8 | 0 | **1.7** | **98.45** | **9** | **0** | 2.8 | 98.45 | 9 | 0 |
| m109 | 2 | 372 | 98.73 | 22 | 0 | 19 | 98.73 | 22 | 0 | **5.9** | **98.73** | **28** | **0** | 7.4 | 98.73 | 28 | 0 |
| m109 | 6 | 299 | 98.73 | 14 | 0 | 6.2 | 98.73 | 14 | 0 | **2.8** | **98.73** | **16** | **0** | 3.1 | 98.73 | 16 | 0 |
| m109 | 10 | 350 | 98.73 | 12 | 0 | 4.7 | 98.73 | 12 | 0 | **2.6** | **98.73** | **16** | **0** | 3.9 | 98.73 | 16 | 0 |
| m110 | 1 | *1801* | *87.01* | 23 | *11.85* | 14 | 87.43 | 24 | 0 | 46 | 87.43 | 29 | 0 | **8.2** | **87.43** | **29** | **0** |
| m110 | 7 | *1800* | *87.57* | *8* | *11.12* | *1801* | *87.57* | *8* | *1.13* | 13 | 87.57 | 10 | 0 | **7.2** | **87.57** | **10** | **0** |
| m110 | 8 | *1802* | *87.43* | *9* | *11.3* | *1801* | *87.57* | *10* | *1.11* | **20** | **87.57** | **18** | **0** | 28 | 87.57 | 18 | 0 |

Table 4.7: Detailed results for extreme point generation for custom instances with 60 assets

| Instance | br | (D-APP) | | | (D-APP) + VI | | | | (D-APP-V) | | | (D-APP-V) + VI | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap | ST | TPV | Dev | Gap |
| m101 | 7 | *1801* | *95.92* | *13* | *0.13* | 22 | 95.92 | 13 | 0 | 21 | 95.92 | 13 | 0 | **3.7** | **95.92** | **13** | **0** |
| m101 | 8 | *1808* | *94.6* | *12* | *1.52* | 103 | 95.92 | 22 | 0 | 12 | 95.92 | 22 | 0 | **6.6** | **95.92** | **22** | **0** |
| m101 | 9 | *1801* | *92.32* | *9* | *4.03* | 39 | 95.92 | 20 | 0 | 10 | 95.92 | 20 | 0 | **4.6** | **95.92** | **20** | **0** |
| m102 | 1 | *1801* | *79.83* | *18* | *21.95* | *1812* | *80.43* | *24* | *10.29* | *1803* | *85.23* | *53* | *2.67* | *1808* | *85.23* | *53* | *0.98* |
| m102 | 9 | *1811* | *80.79* | *8* | *20.5* | *1809* | *83.19* | *20* | *6.92* | **859** | **86.91** | **28** | **0** | *1801* | *86.91* | *28* | *0.26* |
| m102 | 10 | *1811* | *81.15* | *10* | *19.96* | *1801* | *85.59* | *29* | *3.92* | **587** | **86.91** | **28** | **0** | 824 | 86.91 | 28 | 0 |
| m103 | 5 | *1801* | *94.96* | *11* | *5.3* | *1801* | *95.56* | *13* | *4.64* | 650 | 96.4 | 29 | 0 | **595** | **96.4** | **29** | **0** |
| m103 | 7 | *1801* | *99.04* | *15* | *0.97* | 798 | 99.04 | 15 | 0 | 13 | 99.04 | 15 | 0 | **6.5** | **99.04** | **15** | **0** |
| m103 | 10 | *1801* | *97.72* | *14* | *2.33* | *1804* | *97.12* | *11* | *2.59* | **846** | **98.92** | **18** | **0** | 1042 | 98.92 | 18 | 0 |
| m104 | 2 | *1802* | *88.36* | *18* | *11.01* | *1809* | *92.2* | *27* | *6.37* | *1802* | *96.76* | *50* | *1.23* | *1807* | *96.76* | *54* | *1.36* |
| m104 | 4 | *1801* | *84.51* | *10* | *16.05* | *1801* | *94.12* | *44* | *4.08* | 282 | 96.76 | 37 | 0 | **56** | **96.76** | **37** | **0** |
| m104 | 8 | *1802* | *93.76* | *14* | *4.61* | *1801* | *97.84* | *36* | *0.25* | *1804* | *98.08* | *61* | *0.01* | **1617** | **98.08** | **61** | **0** |
| m105 | 4 | *1805* | *94.12* | *26* | *4.21* | 64 | 98.08 | 38 | 0 | **15** | **98.08** | **38** | **0** | 23 | 98.08 | 38 | 0 |
| m105 | 9 | *1801* | *98.08* | *19* | *0.0* | 34 | 98.08 | 19 | 0 | **4.4** | **98.08** | **19** | **0** | 6.5 | 98.08 | 19 | 0 |
| m105 | 10 | *1801* | *98.08* | *19* | *0.0* | 27 | 98.08 | 19 | 0 | **6.2** | **98.08** | **19** | **0** | 7.5 | 98.08 | 19 | 0 |
| m106 | 2 | *1807* | *88.24* | *16* | *13.06* | *1801* | *91.12* | *21* | *6.71* | 597 | 94.96 | 36 | 0 | **159** | **94.96** | **36** | **0** |
| m106 | 3 | *1801* | *91.48* | *25* | *9.06* | *1801* | *91.6* | *26* | *6.15* | **249** | **94.96** | **42** | **0** | 304 | 94.96 | 42 | 0 |
| m106 | 6 | *1801* | *96.52* | *27* | *3.36* | *1801* | *98.2* | *38* | *0.97* | *1801* | *98.44* | *47* | *0.12* | ***1800*** | ***98.44*** | **47** | ***0.11*** |
| m107 | 5 | *1801* | *90.52* | *17* | *8.22* | *1801* | *93.16* | *25* | *4.37* | ***1800*** | ***94.48*** | **34** | ***0.72*** | *1801* | *94.48* | *34* | *0.88* |
| m107 | 6 | *1801* | *97.12* | *17* | *0.87* | 759 | 97.36 | 20 | 0 | **7.7** | **97.36** | **19** | **0** | 13 | 97.36 | 19 | 0 |
| m107 | 9 | *1801* | *92.68* | *18* | *5.7* | *1806* | *97.36* | *34* | *0.61* | 47 | 97.36 | 34 | 0 | **31** | **97.36** | **34** | **0** |
| m108 | 1 | *1801* | *89.08* | *24* | *11.86* | *1801* | *96.76* | *43* | *0.75* | **19** | **96.88** | **49** | **0** | 54 | 96.88 | 49 | 0 |
| m108 | 6 | *1801* | *98.32* | *13* | *1.34* | 9.1 | 98.32 | 13 | 0 | **2.0** | **98.32** | **13** | **0** | 3.7 | 98.32 | 13 | 0 |
| m108 | 9 | *1805* | *95.56* | *21* | *4.27* | *1801* | *96.76* | *22* | *0.74* | **20** | **96.88** | **29** | **0** | 34 | 96.88 | 29 | 0 |
| m109 | 4 | *1801* | *90.04* | *25* | *7.33* | *1801* | *94.12* | *44* | *0.64* | **43** | **94.6** | **43** | **0** | 85 | 94.6 | 43 | 0 |
| m109 | 8 | *1801* | *93.76* | *14* | *3.07* | 1253 | 95.56 | 38 | 0 | 66 | 95.56 | 38 | 0 | **66** | **95.56** | **38** | **0** |
| m109 | 10 | *1801* | *94.96* | *22* | *1.77* | 284 | 95.56 | 25 | 0 | **20** | **95.56** | **25** | **0** | 21 | 95.56 | 25 | 0 |
| m110 | 3 | *1801* | *84.51* | *9* | *15.62* | 573 | 88.84 | 23 | 0 | 32 | 89.08 | 21 | 0 | **6.5** | **89.08** | **21** | **0** |
| m110 | 4 | *1808* | *77.07* | *10* | *27.87* | *1801* | *83.91* | *33* | *4.86* | 1381 | 88.0 | 46 | 0 | **494** | **88.0** | **46** | **0** |
| m110 | 7 | *1815* | *87.27* | *16* | *12.93* | *1801* | *89.2* | *18* | *1.34* | 1198 | 89.2 | 18 | 0 | **158** | **89.2** | **18** | **0** |

Table 4.8: Computational results for extreme point generation for custom instances

| | (D-APP) | | (D-APP) + VI | | (D-APP-V) | | (D-APP-V) + VI | |
|---|---|---|---|---|---|---|---|---|
| n | ST | TPV | ST | TPV | ST | TPV | ST | TPV |
| 30 | 127 | 96.29 (0.5) | 2.88 | 96.29 | 1.13 | 96.29 | 1.35 | 96.29 |
| 40 | 81.2 | 96.02 (1.7) | 105 | 96.4 (0.2) | 9.24 | 96.4 | 7.82 | 96.4 |
| 50 | 465 | 94.31 (4.3) | 199 | 95.29 (0.5) | 78.9 | 95.42 | 43.9 | 95.42 |
| 60 | – | 91.35 (8.0) | 330 | 93.83 (2.2) | 279 | 94.95 (0.2) | 225 | 94.95 (0.1) |

From Table 4.8, we observed that:

- The initial model (D-APP) rapidly reached its limit for the exact solving of our problem. It could be used to solve most of the small instances within less than 2 minutes. For more than 40 assets, we did not obtain good solutions in a reasonable time frame.
- The use of valid inequalities greatly improved formulation (D-APP). We could solve all 30-asset instances to optimality in less than 3 seconds. We solved additional instances for all sizes, yielding near-optimal solutions for instances with 50 assets in less than 200 seconds on average.
- The valid inequalities are not sufficient for reliably solving larger instances using initial formulation (D-APP). We reached the time limit more often for instances with 60 assets, resulting in an average 2.2% optimality gap.
- The reformulation (D-APP-V) we proposed clearly outperformed the initial model. Solving time dropped for every instance, generating the extreme point for all instances up to 40 assets in 10 seconds on average, and for instances with 50 assets in 80 seconds.
- Adding the valid inequalities had a lesser impact on the reformulation. For instances with 30 and 40 assets, the time necessary to compute the valid inequalities can slightly outweigh the solution time reduction. However, when this computation time becomes negligible, we reduced the average solution time by 36 seconds for instances with 50 assets, and 54 seconds for instances with 60 assets.

### 4.5.3 *Impact analysis of valid inequalities*

Since initial model (D-APP) was not able to retrieve the first extreme point within the time limit for most instances with 50 or more assets, we can therefore not use the model to generate the full Pareto front within a reasonable time.

We generated the Pareto fronts using reformulation (D-APP-V) with different combinations of the valid inequalities presented in Section 4.4. We divided the valid inequalities into three groups:

- Deviation-based inequalities (DB): (4.30) + (4.31) + (4.32) + (4.39)

- Incompatibility/vehicle clique inequalities (IV): (4.43) + (4.44) + (4.45)
- Clique protection inequalities (CP): (3.25)

We compared the model with no valid inequality with each pair of the valid inequality groups, and with all the valid inequalities. As in Section 4.5.2, we considered each of our ten instances, with three different random vehicle breakdowns, and a time limit of 1800 seconds.

Detailed results for custom instances of size 30, 40, 50 and 60 are given in Tables 4.9, 4.10, 4.11 and 4.12, respectively. For each instance, column "br" shows the vehicle that broke down. For every combination of valid inequalities, column "ST" shows the solution time in seconds, column "Nb" indicates the number of points in the front, and column "HV" gives the hypervolume of the front. The last two "Optimal" columns show the number of points and the hypervolume of the optimal Pareto front, when it is known.

We can see that no combination of the valid inequalities dominates another. The average number of points in the optimal Pareto front and the solution time increases as the number of assets goes from 30 to 50. The number of points in the optimal Pareto front are similar for larger instances, ranging from 9 to 28 for instances with 50 and from 9 to 38 for instances with 60 assets. However, points for instances with 60 assets are harder to compute, increasing the solution time.

Table 4.13 presents the aggregated results for the generation of the Pareto front using (D-APP-V), with and without valid inequalities. Column "ST" shows the average solution time to obtain the Pareto front. Column "#Opt" shows the number of instances for which the Pareto front is obtained within the time limit.

Table 4.9: Detailed results for Pareto front generation for custom instances with 30 assets

| Instance | br | None | | | (DB) + (IV) | | | (DB) + (CP) | | | (IV) + (CP) | | | All | | | Optimal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | Nb | HV |
| m101 | 1 | 7.1 | 9 | 61.0 | 3.6 | 9 | 61.0 | 2.9 | 9 | 61.0 | 2.5 | 9 | 61.0 | 3.2 | 9 | 61.0 | 9 | 61.0 |
| m101 | 3 | 4.6 | 15 | 64.3 | 5.7 | 15 | 64.3 | 5.1 | 15 | 64.3 | 5.1 | 15 | 64.3 | 5.8 | 15 | 64.3 | 15 | 64.3 |
| m101 | 9 | 1.4 | 7 | 60.0 | 2.4 | 7 | 60.0 | 2.3 | 7 | 60.0 | 1.7 | 7 | 60.0 | 2.5 | 7 | 60.0 | 7 | 60.0 |
| m102 | 1 | 2.4 | 7 | 77.4 | 3.3 | 7 | 77.4 | 3.1 | 7 | 77.4 | 2.7 | 7 | 77.4 | 3.5 | 7 | 77.4 | 7 | 77.4 |
| m102 | 4 | 2.2 | 10 | 55.9 | 3.1 | 10 | 55.9 | 3.1 | 10 | 55.9 | 2.8 | 10 | 55.9 | 3.0 | 10 | 55.9 | 10 | 55.9 |
| m102 | 8 | 3.3 | 9 | 75.6 | 4.0 | 9 | 75.6 | 3.9 | 9 | 75.6 | 3.2 | 9 | 75.6 | 4.1 | 9 | 75.6 | 9 | 75.6 |
| m103 | 5 | 2.5 | 9 | 63.5 | 3.2 | 9 | 63.5 | 3.3 | 9 | 63.5 | 2.2 | 9 | 63.5 | 3.1 | 9 | 63.5 | 9 | 63.5 |
| m103 | 6 | 1.7 | 6 | 65.4 | 2.4 | 6 | 65.4 | 2.2 | 6 | 65.4 | 1.8 | 6 | 65.4 | 2.5 | 6 | 65.4 | 6 | 65.4 |
| m103 | 10 | 0.68 | 2 | 0.0 | 0.92 | 2 | 0.0 | 0.94 | 2 | 0.0 | 0.57 | 2 | 0.0 | 0.94 | 2 | 0.0 | 2 | 0.0 |
| m104 | 1 | 4.2 | 8 | 68.3 | 5.2 | 8 | 68.3 | 5.5 | 8 | 68.3 | 4.2 | 8 | 68.3 | 5.4 | 8 | 68.3 | 8 | 68.3 |
| m104 | 9 | 4.7 | 10 | 67.5 | 5.7 | 10 | 67.5 | 5.9 | 10 | 67.5 | 5.7 | 10 | 67.5 | 5.4 | 10 | 67.5 | 10 | 67.5 |
| m104 | 5 | 9.0 | 13 | 65.3 | 7.7 | 13 | 65.3 | 8.2 | 13 | 65.3 | 7.2 | 13 | 65.3 | 7.9 | 13 | 65.3 | 13 | 65.3 |
| m105 | 6 | 1.7 | 8 | 61.3 | 2.6 | 8 | 61.3 | 2.7 | 8 | 61.3 | 2.0 | 8 | 61.3 | 3.8 | 8 | 61.3 | 8 | 61.3 |
| m105 | 8 | 1.7 | 4 | 37.2 | 3.6 | 4 | 37.2 | 3.2 | 4 | 37.2 | 2.1 | 4 | 37.2 | 3.8 | 4 | 37.2 | 4 | 37.2 |
| m105 | 10 | 2.7 | 5 | 49.4 | 2.1 | 5 | 49.4 | 2.3 | 5 | 49.4 | 1.4 | 5 | 49.4 | 2.1 | 5 | 49.4 | 5 | 49.4 |
| m106 | 3 | 9.1 | 11 | 59.8 | 11 | 11 | 59.8 | 8.2 | 11 | 59.8 | 8.7 | 11 | 59.8 | 9.6 | 11 | 59.8 | 11 | 59.8 |
| m106 | 5 | 7.8 | 15 | 73.4 | 7.1 | 15 | 73.4 | 7.2 | 15 | 73.4 | 6.4 | 15 | 73.4 | 7.9 | 15 | 73.4 | 15 | 73.4 |
| m106 | 8 | 4.2 | 7 | 45.7 | 2.7 | 7 | 45.7 | 2.6 | 7 | 45.7 | 2.0 | 7 | 45.7 | 2.6 | 7 | 45.7 | 7 | 45.7 |
| m107 | 1 | 2.5 | 11 | 66.3 | 3.6 | 11 | 66.3 | 3.1 | 11 | 66.3 | 3.0 | 11 | 66.3 | 3.7 | 11 | 66.3 | 11 | 66.3 |
| m107 | 3 | 2.2 | 8 | 69.9 | 2.9 | 8 | 69.9 | 2.9 | 8 | 69.9 | 2.5 | 8 | 69.9 | 3.1 | 8 | 69.9 | 8 | 69.9 |
| m107 | 10 | 4.1 | 12 | 64.9 | 5.4 | 12 | 64.9 | 5.0 | 12 | 64.9 | 4.3 | 12 | 64.9 | 5.3 | 12 | 64.9 | 12 | 64.9 |
| m108 | 2 | 3.6 | 9 | 73.7 | 4.7 | 9 | 73.7 | 4.3 | 9 | 73.7 | 3.7 | 9 | 73.7 | 4.6 | 9 | 73.7 | 9 | 73.7 |
| m108 | 4 | 2.2 | 10 | 60.5 | 3.3 | 10 | 60.5 | 2.9 | 10 | 60.5 | 2.4 | 10 | 60.5 | 3.3 | 10 | 60.5 | 10 | 60.5 |
| m108 | 5 | 1.2 | 6 | 40.9 | 2.0 | 6 | 40.9 | 2.0 | 6 | 40.9 | 1.5 | 6 | 40.9 | 2.1 | 6 | 40.9 | 6 | 40.9 |
| m109 | 1 | 5.9 | 16 | 68.8 | 7.9 | 16 | 68.8 | 7.6 | 16 | 68.8 | 7.4 | 16 | 68.8 | 8.2 | 16 | 68.8 | 16 | 68.8 |
| m109 | 4 | 3.2 | 9 | 69.1 | 3.8 | 9 | 69.1 | 3.2 | 9 | 69.1 | 3.0 | 9 | 69.1 | 3.5 | 9 | 69.1 | 9 | 69.1 |
| m109 | 10 | 4.5 | 10 | 70.6 | 5.3 | 10 | 70.6 | 5.4 | 10 | 70.6 | 3.8 | 10 | 70.6 | 5.3 | 10 | 70.6 | 10 | 70.6 |
| m110 | 4 | 3.4 | 10 | 72.5 | 3.2 | 10 | 72.5 | 3.7 | 10 | 72.5 | 2.9 | 10 | 72.5 | 3.2 | 10 | 72.5 | 10 | 72.5 |
| m110 | 7 | 3.2 | 9 | 59.1 | 2.9 | 9 | 59.1 | 3.8 | 9 | 59.1 | 2.7 | 9 | 59.1 | 2.9 | 9 | 59.1 | 9 | 59.1 |
| m110 | 9 | 1.9 | 8 | 59.1 | 2.5 | 8 | 59.1 | 2.7 | 8 | 59.1 | 1.9 | 8 | 59.1 | 2.6 | 8 | 59.1 | 8 | 59.1 |

Table 4.10: Detailed results for Pareto front generation for custom instances with 40 assets

| Instance | br | None | | | (DB) + (IV) | | | (DB) + (CP) | | | (IV) + (CP) | | | All | | | Optimal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | Nb | HV |
| m101 | 2 | 13 | 12 | 73.7 | 12 | 12 | 73.7 | 13 | 12 | 73.7 | 15 | 12 | 73.7 | 13 | 12 | 73.7 | 12 | 73.7 |
| m101 | 3 | 13 | 12 | 70.3 | 15 | 12 | 70.3 | 12 | 12 | 70.3 | 12 | 12 | 70.3 | 14 | 12 | 70.3 | 12 | 70.3 |
| m101 | 4 | 9.3 | 15 | 68.8 | 13 | 15 | 68.8 | 11 | 15 | 68.8 | 9.1 | 15 | 68.8 | 13 | 15 | 68.8 | 15 | 68.8 |
| m102 | 2 | 351 | 18 | 70.0 | 623 | 18 | 70.0 | 547 | 18 | 70.0 | *900* | 3 | 43.2 | 621 | 18 | 70.0 | 18 | 70.0 |
| m102 | 8 | 95 | 15 | 61.3 | 44 | 15 | 61.3 | 27 | 15 | 61.3 | 71 | 15 | 61.3 | 44 | 15 | 61.3 | 15 | 61.3 |
| m102 | 10 | 62 | 15 | 75.1 | 77 | 15 | 75.1 | 52 | 15 | 75.1 | 66 | 15 | 75.1 | 74 | 15 | 75.1 | 15 | 75.1 |
| m103 | 5 | 9.6 | 11 | 58.1 | 8.9 | 11 | 58.1 | 12 | 11 | 58.1 | 10 | 11 | 58.1 | 8.6 | 11 | 58.1 | 11 | 58.1 |
| m103 | 6 | 5.5 | 13 | 67.8 | 7.4 | 13 | 67.8 | 6.6 | 13 | 67.8 | 6.4 | 13 | 67.8 | 7.7 | 13 | 67.8 | 13 | 67.8 |
| m103 | 9 | 4.6 | 9 | 69.3 | 6.8 | 9 | 69.3 | 5.9 | 9 | 69.3 | 5.5 | 9 | 69.3 | 6.9 | 9 | 69.3 | 9 | 69.3 |
| m104 | 4 | 28 | 18 | 68.4 | 34 | 18 | 68.4 | 27 | 18 | 68.4 | 32 | 18 | 68.4 | 34 | 18 | 68.4 | 18 | 68.4 |
| m104 | 5 | 95 | 26 | 68.1 | 67 | 26 | 68.1 | 66 | 26 | 68.1 | 76 | 26 | 68.1 | 66 | 26 | 68.1 | 26 | 68.1 |
| m104 | 8 | 4.3 | 8 | 55.0 | 5.8 | 8 | 55.0 | 6.0 | 8 | 55.0 | 5.1 | 8 | 55.0 | 6.2 | 8 | 55.0 | 8 | 55.0 |
| m105 | 1 | 4.5 | 12 | 65.5 | 6.5 | 12 | 65.5 | 6.2 | 12 | 65.5 | 4.5 | 12 | 65.5 | 6.8 | 12 | 65.5 | 12 | 65.5 |
| m105 | 7 | 2.6 | 7 | 53.4 | 4.4 | 7 | 53.4 | 4.2 | 7 | 53.4 | 2.8 | 7 | 53.4 | 4.4 | 7 | 53.4 | 7 | 53.4 |
| m105 | 10 | 2.2 | 6 | 49.1 | 3.7 | 6 | 49.1 | 3.4 | 6 | 49.1 | 2.4 | 6 | 49.1 | 3.4 | 6 | 49.1 | 6 | 49.1 |
| m106 | 4 | 21 | 15 | 58.7 | 22 | 15 | 58.7 | 18 | 15 | 58.7 | 23 | 15 | 58.7 | 24 | 15 | 58.7 | 15 | 58.7 |
| m106 | 7 | 14 | 10 | 58.0 | 9.7 | 10 | 58.0 | 8.8 | 10 | 58.0 | 13 | 10 | 58.0 | 11 | 10 | 58.0 | 10 | 58.0 |
| m106 | 10 | 12 | 15 | 65.4 | 20 | 15 | 65.4 | 14 | 15 | 65.4 | 22 | 15 | 65.4 | 18 | 15 | 65.4 | 15 | 65.4 |
| m107 | 3 | 17 | 12 | 73.2 | 18 | 12 | 73.2 | 15 | 12 | 73.2 | 20 | 12 | 73.2 | 19 | 12 | 73.2 | 12 | 73.2 |
| m107 | 8 | 28 | 15 | 70.6 | 27 | 15 | 70.6 | 23 | 15 | 70.6 | 24 | 15 | 70.6 | 29 | 15 | 70.6 | 15 | 70.6 |
| m107 | 10 | 126 | 22 | 71.6 | 110 | 22 | 71.6 | 92 | 22 | 71.6 | 137 | 22 | 71.6 | 109 | 22 | 71.6 | 22 | 71.6 |
| m108 | 2 | 11 | 16 | 59.6 | 11 | 16 | 59.6 | 14 | 16 | 59.6 | 9.4 | 16 | 59.6 | 11 | 16 | 59.6 | 16 | 59.6 |
| m108 | 6 | 4.8 | 11 | 54.1 | 7.1 | 11 | 54.1 | 6.4 | 11 | 54.1 | 5.0 | 11 | 54.1 | 6.8 | 11 | 54.1 | 11 | 54.1 |
| m108 | 7 | 1.8 | 4 | 44.2 | 3.1 | 4 | 44.2 | 2.8 | 4 | 44.2 | 2.1 | 4 | 44.2 | 3.1 | 4 | 44.2 | 4 | 44.2 |
| m109 | 6 | 5.7 | 8 | 72.3 | 6.5 | 8 | 72.3 | 6.3 | 8 | 72.3 | 6.6 | 8 | 72.3 | 6.8 | 8 | 72.3 | 8 | 72.3 |
| m109 | 7 | 4.6 | 8 | 64.5 | 6.0 | 8 | 64.5 | 6.3 | 8 | 64.5 | 5.3 | 8 | 64.5 | 6.0 | 8 | 64.5 | 8 | 64.5 |
| m109 | 9 | 6.6 | 11 | 67.2 | 7.0 | 11 | 67.2 | 7.6 | 11 | 67.2 | 6.4 | 11 | 67.2 | 7.0 | 11 | 67.2 | 11 | 67.2 |
| m110 | 5 | 4.2 | 7 | 58.1 | 4.1 | 7 | 58.1 | 4.1 | 7 | 58.1 | 3.0 | 7 | 58.1 | 4.1 | 7 | 58.1 | 7 | 58.1 |
| m110 | 6 | 23 | 10 | 75.0 | 9.0 | 10 | 75.0 | 25 | 10 | 75.0 | 13 | 10 | 75.0 | 9.5 | 10 | 75.0 | 10 | 75.0 |
| m110 | 7 | 109 | 21 | 74.2 | 60 | 21 | 74.2 | 104 | 21 | 74.2 | 56 | 21 | 74.2 | 62 | 21 | 74.2 | 21 | 74.2 |

Table 4.11: Detailed results for Pareto front generation for custom instances with 50 assets

| Instance | br | None | | | (DB) + (IV) | | | (DB) + (CP) | | | (IV) + (CP) | | | All | | | Optimal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | Nb | HV |
| m101 | 3 | 67 | 23 | 72.6 | 79 | 23 | 72.6 | 66 | 23 | 72.6 | 70 | 23 | 72.6 | 87 | 23 | 72.6 | 23 | 72.6 |
| m101 | 4 | 66 | 17 | 74.1 | 64 | 17 | 74.1 | 52 | 17 | 74.1 | 59 | 17 | 74.1 | 56 | 17 | 74.1 | 17 | 74.1 |
| m101 | 6 | 10 | 13 | 67.7 | 23 | 13 | 67.7 | 13 | 13 | 67.7 | 10 | 13 | 67.7 | 14 | 13 | 67.7 | 13 | 67.7 |
| m102 | 6 | 824 | 16 | 39.6 | 556 | 16 | 39.6 | 794 | 16 | 39.6 | 949 | 16 | 39.6 | 553 | 16 | 39.6 | 16 | 39.6 |
| m102 | 8 | 780 | 9 | 48.3 | 782 | 9 | 48.3 | 839 | 9 | 48.3 | 694 | 9 | 48.3 | 787 | 9 | 48.3 | 9 | 48.3 |
| m102 | 10 | 280 | 17 | 51.6 | 131 | 17 | 51.6 | 113 | 17 | 51.6 | 248 | 17 | 51.6 | 131 | 17 | 51.6 | 17 | 51.6 |
| m103 | 2 | 38 | 18 | 64.0 | 45 | 18 | 64.0 | 39 | 18 | 64.0 | 46 | 18 | 64.0 | 43 | 18 | 64.0 | 18 | 64.0 |
| m103 | 7 | 10 | 10 | 62.3 | 13 | 10 | 62.3 | 13 | 10 | 62.3 | 12 | 10 | 62.3 | 14 | 10 | 62.3 | 10 | 62.3 |
| m103 | 9 | 32 | 16 | 69.2 | 34 | 16 | 69.2 | 29 | 16 | 69.2 | 31 | 16 | 69.2 | 35 | 16 | 69.2 | 16 | 69.2 |
| m104 | 3 | 61 | 19 | 84.5 | 46 | 19 | 84.5 | 55 | 19 | 84.5 | 70 | 19 | 84.5 | 48 | 19 | 84.5 | 19 | 84.5 |
| m104 | 4 | 52 | 17 | 65.9 | 54 | 17 | 65.9 | 33 | 17 | 65.9 | 46 | 17 | 65.9 | 40 | 17 | 65.9 | 17 | 65.9 |
| m104 | 5 | 138 | 17 | 60.5 | 50 | 17 | 60.5 | 48 | 17 | 60.5 | 47 | 17 | 60.5 | 47 | 17 | 60.5 | 17 | 60.5 |
| m105 | 3 | 19 | 19 | 67.1 | 20 | 19 | 67.1 | 21 | 19 | 67.1 | 17 | 19 | 67.1 | 19 | 19 | 67.1 | 19 | 67.1 |
| m105 | 5 | 15 | 13 | 77.3 | 17 | 13 | 77.3 | 16 | 13 | 77.3 | 16 | 13 | 77.3 | 19 | 13 | 77.3 | 13 | 77.3 |
| m105 | 10 | 10 | 11 | 60.4 | 13 | 11 | 60.4 | 12 | 11 | 60.4 | 9.7 | 11 | 60.4 | 13 | 11 | 60.4 | 11 | 60.4 |
| m106 | 1 | 46 | 19 | 63.1 | 60 | 19 | 63.1 | 52 | 19 | 63.1 | 52 | 19 | 63.1 | 49 | 19 | 63.1 | 19 | 63.1 |
| m106 | 4 | 57 | 26 | 57.0 | 56 | 26 | 57.0 | 51 | 26 | 57.0 | 58 | 26 | 57.0 | 60 | 26 | 57.0 | 26 | 57.0 |
| m106 | 7 | 77 | 18 | 73.4 | 71 | 18 | 73.4 | 50 | 18 | 73.4 | 88 | 18 | 73.4 | 58 | 18 | 73.4 | 18 | 73.4 |
| m107 | 1 | 408 | 28 | 72.6 | 439 | 28 | 72.6 | 325 | 28 | 72.6 | 494 | 28 | 72.6 | 444 | 28 | 72.6 | 28 | 72.6 |
| m107 | 7 | 17 | 10 | 63.0 | 14 | 10 | 63.0 | 15 | 10 | 63.0 | 11 | 10 | 63.0 | 15 | 10 | 63.0 | 10 | 63.0 |
| m107 | 9 | 29 | 9 | 72.1 | 46 | 9 | 72.1 | 24 | 9 | 72.1 | 76 | 9 | 72.1 | 46 | 9 | 72.1 | 9 | 72.1 |
| m108 | 1 | 26 | 19 | 65.5 | 35 | 19 | 65.5 | 29 | 19 | 65.5 | 26 | 19 | 65.5 | 33 | 19 | 65.5 | 19 | 65.5 |
| m108 | 7 | 9.2 | 11 | 63.1 | 12 | 11 | 63.1 | 12 | 11 | 63.1 | 10 | 11 | 63.1 | 14 | 11 | 63.1 | 11 | 63.1 |
| m108 | 8 | 9.4 | 10 | 60.9 | 13 | 10 | 60.9 | 11 | 10 | 60.9 | 10 | 10 | 60.9 | 12 | 10 | 60.9 | 10 | 60.9 |
| m109 | 2 | 62 | 18 | 80.6 | 60 | 18 | 80.6 | 64 | 18 | 80.6 | 64 | 18 | 80.6 | 59 | 18 | 80.6 | 18 | 80.6 |
| m109 | 6 | 16 | 14 | 73.5 | 19 | 14 | 73.5 | 19 | 14 | 73.5 | 19 | 14 | 73.5 | 20 | 14 | 73.5 | 14 | 73.5 |
| m109 | 10 | 23 | 15 | 75.5 | 29 | 15 | 75.5 | 27 | 15 | 75.5 | 28 | 15 | 75.5 | 32 | 15 | 75.5 | 15 | 75.5 |
| m110 | 1 | 231 | 21 | 64.0 | 104 | 21 | 64.0 | 103 | 21 | 64.0 | 89 | 21 | 64.0 | 80 | 21 | 64.0 | 21 | 64.0 |
| m110 | 7 | 25 | 10 | 49.7 | 19 | 10 | 49.7 | 23 | 10 | 49.7 | 21 | 10 | 49.7 | 19 | 10 | 49.7 | 10 | 49.7 |
| m110 | 8 | 73 | 16 | 57.7 | 67 | 16 | 57.7 | 64 | 16 | 57.7 | 77 | 16 | 57.7 | 71 | 16 | 57.7 | 16 | 57.7 |

Table 4.12: Detailed results for Pareto front generation for custom instances with 60 assets

| Instance | br | None | | | (DB) + (IV) | | | (DB) + (CP) | | | (IV) + (CP) | | | All | | | Optimal | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | ST | Nb | HV | Nb | HV |
| m101 | 7 | 37 | 10 | 70.3 | 22 | 10 | 70.3 | 20 | 10 | 70.3 | 14 | 10 | 70.3 | 19 | 10 | 70.3 | 10 | 70.3 |
| m101 | 8 | 88 | 14 | 78.3 | 57 | 14 | 78.3 | 57 | 14 | 78.3 | 38 | 14 | 78.3 | 41 | 14 | 78.3 | 14 | 78.3 |
| m101 | 9 | 73 | 17 | 71.1 | 39 | 17 | 71.1 | 46 | 17 | 71.1 | 30 | 17 | 71.1 | 37 | 17 | 71.1 | 17 | 71.1 |
| m102 | 1 | *1802* | 1 | 2.3 | *1804* | 1 | 2.3 | *1800* | 1 | 2.3 | *1801* | 1 | 0.0 | *1803* | 1 | 2.3 | – | – |
| m102 | 9 | 1037 | 18 | 49.7 | *1800* | 1 | 0.0 | 1343 | 18 | 49.7 | 908 | 18 | 49.7 | *1800* | 1 | 0.0 | 18 | 49.7 |
| m102 | 10 | 791 | 23 | 47.3 | 910 | 23 | 47.3 | 576 | 23 | 47.3 | 1636 | 23 | 47.3 | 914 | 23 | 47.3 | 23 | 47.3 |
| m103 | 5 | 1567 | 18 | 63.3 | *1802* | 10 | 49.6 | 1118 | 18 | 63.3 | *1800* | 1 | 0.0 | *1800* | 15 | 61.1 | 18 | 63.3 |
| m103 | 7 | 49 | 15 | 64.8 | 39 | 15 | 64.8 | 42 | 15 | 64.8 | 37 | 15 | 64.8 | 43 | 15 | 64.8 | 15 | 64.8 |
| m103 | 10 | 968 | 15 | 66.5 | 1187 | 15 | 66.5 | 613 | 15 | 66.5 | 1600 | 15 | 66.5 | 1179 | 15 | 66.5 | 15 | 66.5 |
| m104 | 2 | *1803* | 1 | 6.9 | *1805* | 1 | 6.9 | *1802* | 1 | 6.9 | *1804* | 1 | 6.9 | *1805* | 1 | 0.0 | – | – |
| m104 | 4 | 754 | 24 | 73.7 | 325 | 24 | 73.7 | 1025 | 24 | 73.7 | 431 | 24 | 73.7 | 244 | 24 | 73.7 | 24 | 73.7 |
| m104 | 8 | *1802* | 1 | 3.2 | *1800* | 2 | 30.0 | *1801* | 1 | 3.2 | *1802* | 1 | 0.0 | *1800* | 2 | 30.0 | – | – |
| m105 | 4 | 151 | 26 | 78.7 | 179 | 26 | 78.7 | 159 | 26 | 78.7 | 172 | 26 | 78.7 | 181 | 26 | 78.7 | 26 | 78.7 |
| m105 | 9 | 40 | 18 | 65.8 | 46 | 18 | 65.8 | 44 | 18 | 65.8 | 43 | 18 | 65.8 | 46 | 18 | 65.8 | 18 | 65.8 |
| m105 | 10 | 45 | 16 | 70.8 | 50 | 16 | 70.8 | 45 | 16 | 70.8 | 45 | 16 | 70.8 | 51 | 16 | 70.8 | 16 | 70.8 |
| m106 | 2 | 1585 | 32 | 65.1 | *1800* | 20 | 52.0 | 923 | 32 | 65.1 | *1800* | 17 | 46.4 | 1453 | 32 | 65.1 | 32 | 65.1 |
| m106 | 3 | 1440 | 33 | 68.5 | 1552 | 33 | 68.5 | 1040 | 33 | 68.5 | *1800* | 9 | 32.4 | 1559 | 33 | 68.5 | 33 | 68.5 |
| m106 | 6 | *1801* | 1 | 0.0 | *1800* | 1 | 0.0 | *1801* | 1 | 0.0 | *1802* | 1 | 0.0 | *1800* | 1 | 0.0 | – | – |
| m107 | 5 | *1800* | 1 | 0.0 | *1800* | 1 | 0.0 | *1800* | 4 | 20.5 | *1800* | 1 | 0.0 | *1801* | 1 | 0.0 | – | – |
| m107 | 6 | 37 | 15 | 72.0 | 44 | 15 | 72.0 | 35 | 15 | 72.0 | 33 | 15 | 72.0 | 39 | 15 | 72.0 | 15 | 72.0 |
| m107 | 9 | 315 | 27 | 76.2 | 260 | 27 | 76.2 | 203 | 27 | 76.2 | 394 | 27 | 76.2 | 260 | 27 | 76.2 | 27 | 76.2 |
| m108 | 1 | 265 | 38 | 77.9 | 302 | 38 | 77.9 | 268 | 38 | 77.9 | 349 | 38 | 77.9 | 357 | 38 | 77.9 | 38 | 77.9 |
| m108 | 6 | 17 | 14 | 65.4 | 22 | 14 | 65.4 | 19 | 14 | 65.4 | 15 | 14 | 65.4 | 21 | 14 | 65.4 | 14 | 65.4 |
| m108 | 9 | 86 | 18 | 76.5 | 99 | 18 | 76.5 | 74 | 18 | 76.5 | 92 | 18 | 76.5 | 110 | 18 | 76.5 | 18 | 76.5 |
| m109 | 4 | 499 | 38 | 75.8 | 425 | 38 | 75.8 | 453 | 38 | 75.8 | 445 | 38 | 75.8 | 441 | 38 | 75.8 | 38 | 75.8 |
| m109 | 8 | 644 | 18 | 85.7 | 496 | 18 | 85.7 | 415 | 18 | 85.7 | 765 | 18 | 85.7 | 503 | 18 | 85.7 | 18 | 85.7 |
| m109 | 10 | 135 | 16 | 81.2 | 128 | 16 | 81.2 | 116 | 16 | 81.2 | 119 | 16 | 81.2 | 128 | 16 | 81.2 | 16 | 81.2 |
| m110 | 3 | 200 | 20 | 45.6 | 99 | 20 | 45.6 | 97 | 20 | 45.6 | 101 | 20 | 45.6 | 102 | 20 | 45.6 | 20 | 45.6 |
| m110 | 4 | *1805* | 2 | 22.5 | 1719 | 30 | 63.3 | 1108 | 30 | 63.3 | *1800* | 10 | 42.3 | 1752 | 30 | 63.3 | 30 | 63.3 |
| m110 | 7 | 1437 | 18 | 52.1 | 305 | 18 | 52.1 | 418 | 18 | 52.1 | 371 | 18 | 52.1 | 353 | 18 | 52.1 | 18 | 52.1 |

Table 4.13: Computational results for Pareto front generation for custom instances

| | None | | (DB) + (IV) | | (DB) + (CP) | | (IV) + (CP) | | All | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ST | #Opt | ST | #Opt | ST | #Opt | ST | #Opt | ST | #Opt |
| n=30 | 4 | 30/30 | 4 | 30/30 | 4 | 30/30 | 3 | 30/30 | 4 | 30/30 |
| 40 | 33 | 30/30 | 28 | 30/30 | 25 | 30/30 | 32 | 30/30 | 29 | 30/30 |
| 50 | 117 | 30/30 | 99 | 30/30 | 100 | 30/30 | 115 | 30/30 | 97 | 30/30 |
| 60 | 511 | 24/30 | 378 | 22/30 | 410 | 25/30 | 364 | 21/30 | 427 | 23/30 |

From Table 4.13, we observed that:

- The valid inequalities we presented further improved the reformulation (D-APP-V).
- For the small custom instances, the model with no valid inequalities can be faster due to the time necessary to build the model. For 30 assets, the computations of bounds for the valid inequalities in group (DB) take approximately 0.5 seconds, and the computation of the cliques takes around 0.1 seconds. For instances with a larger number of assets, the computation times only increase by 1 or 2 seconds, becoming negligible.
- We obtained the optimal Pareto front for all custom instances with up to 50 assets with any combination of the valid inequalities. The (DB) inequalities seemed to always have a positive impact on solution time and to perform better when associated with (CP) inequalities or both (IV) and (CP).
- For instances with 60 assets, the impact of the valid inequalities was less uniform. Depending on the instances, some combinations worked better than others, with no clear dominance. Models solving less instances are expected to have a lower solution time, because the additional solved instances are supposedly harder instances.
- On average, the best combination was valid inequalities (DB) and (CP) that could solve more instances than the model using no valid inequalities and with all valid inequalities – in both cases, faster.

### 4.5.4 *Results for Solomon instances*

As described in Chapter 3, our valid inequalities enabled us to consistently generate initial routes for Solomon instances with 35 assets. Consequently, we can now use these instances in a disrupted setting, which was not previously possible. In the following, we show the impact of the reformulation and valid inequalities when solving the D-APP on Solomon instances and provide results for the complete front generation for future reference. For each instance, we selected a random vehicle breakdown as disruption.

Table 4.14 presents a comparison of the extreme points obtained using (D-APP) and (D-APP-V) with valid inequalities. Column "ST" shows the average solution time to

obtain an optimal solution in seconds. A dash indicates that none of the instances were solved to optimality within the 1800-second time limit. Column "TPV" shows the average total protected value of the feasible solutions obtained at the time limit, with the average relative optimality gap displayed in brackets.

Table 4.14: Computational results for extreme point generation for extended Solomon instances with 35 assets

| | | (D-APP) | | (D-APP-V) + VI | |
|---|---|---|---|---|---|
| Class | Vehicles | ST | TPV | ST | TPV |
| C1 | Set1 | – | 65.64 (46.4) | 147 | 69.26 (0.8) |
| | Set2 | – | 81.2 (22.9) | 594 | 84.3 (3.2) |
| R1 | Set1 | – | 58.42 (55.3) | 226 | 70.59 (0.1) |
| | Set2 | – | 81.0 (22.9) | 512 | 85.17 (2.5) |
| RC1 | Set1 | – | 75.51 (30.3) | 472 | 78.99 |
| | Set2 | – | 92.8 (7.6) | 239 | 93.97 (0.7) |

The results obtained for the Solomon instances were similar to the outcomes for large custom instances. None of these instances could be solved within the time limit using the initial formulation, resulting in high relative optimality gaps.

The reformulation and valid inequalities significantly enhanced the total protected value for all classes of instances and both sets of vehicles. The upper bound was also significantly improved, reducing the relative optimality gaps from between 8% and 55% to less than 3.2% on average.

Table 4.15 provides detailed results for computing the Pareto front using the extended Solomon instances with reformulation (D-APP-V) and all valid inequalities. A time limit of 1800 seconds was set for each instance. For each instance, column "br" shows the vehicle that broke down. Column "ST" shows the solution time in seconds, column "Nb" indicates the number of points in the front, and column "HV" gives the hypervolume of the front.

Table 4.16 presents the aggregated results for Pareto front generation using the reformulation (D-APP-V) with all valid inequalities. Column "ST" shows the average solution time to obtain the optimal Pareto front in seconds. Column "#Opt" shows the number of instances for which we know the optimal Pareto front.

Table 4.15: Detailed results for Pareto front generation for extended Solomon instances with 35 assets

(a) Set1 (9 vehicles)

| Instance | br | (D-APP-V) + VI | | |
|---|---|---|---|---|
| | | ST | Nb | HV |
| 200c101 | 2 | 460 | 6 | 60.3 |
| 200c102 | 5 | 325 | 7 | 47.6 |
| 200c103 | 2 | *1800* | *1* | – |
| 200c104 | 8 | 182 | 5 | 59.5 |
| 200c105 | 9 | 30 | 9 | 40.3 |
| 200c106 | 8 | 12 | 5 | 53.3 |
| 200c107 | 4 | 134 | 6 | 44.4 |
| 200c108 | 2 | *1801* | *1* | – |
| 200c109 | 8 | 18 | 3 | 44.4 |
| 200c110 | 4 | 146 | 5 | 59.3 |
| 200r101 | 5 | 686 | 6 | 63.5 |
| 200r102 | 3 | 649 | 7 | 51.6 |
| 200r103 | 9 | 4.5 | 10 | 57.8 |
| 200r104 | 2 | 44 | 5 | 64.2 |
| 200r105 | 7 | *1800* | *1* | – |
| 200r106 | 5 | 589 | 15 | 72.8 |
| 200r107 | 2 | 54 | 10 | 79.1 |
| 200r108 | 2 | 516 | 11 | 57.9 |
| 200r109 | 5 | 581 | 11 | 76.8 |
| 200r110 | 7 | 19 | 7 | 63.8 |
| 200rc101 | 3 | *1800* | *2* | *0.0* |
| 200rc102 | 6 | 419 | 7 | 73.6 |
| 200rc103 | 6 | 950 | 12 | 76.4 |
| 200rc104 | 7 | 62 | 7 | 56.0 |
| 200rc105 | 2 | 557 | 8 | 63.8 |
| 200rc106 | 9 | 89 | 6 | 47.8 |
| 200rc107 | 8 | 26 | 5 | 58.3 |
| 200rc108 | 2 | 1796 | 13 | 70.1 |
| 200rc109 | 5 | 287 | 4 | 49.0 |
| 200rc110 | 9 | 43 | 9 | 72.5 |

(b) Set2 (12 vehicles)

| Instance | br | (D-APP-V) + VI | | |
|---|---|---|---|---|
| | | ST | Nb | HV |
| 200c101 | 11 | 116 | 8 | 55.0 |
| 200c102 | 11 | 895 | 5 | 60.0 |
| 200c103 | 6 | *1800* | *1* | – |
| 200c104 | 5 | *1801* | *1* | – |
| 200c105 | 1 | *1801* | *1* | – |
| 200c106 | 3 | *1801* | *1* | – |
| 200c107 | 8 | 930 | 8 | 65.7 |
| 200c108 | 3 | *1806* | *1* | – |
| 200c109 | 1 | *1801* | *1* | – |
| 200c110 | 10 | *1803* | *1* | – |
| 200r101 | 10 | *1801* | *1* | – |
| 200r102 | 5 | *1802* | *1* | – |
| 200r103 | 11 | 84 | 10 | 65.6 |
| 200r104 | 6 | 1337 | 8 | 79.5 |
| 200r105 | 2 | *1800* | *1* | – |
| 200r106 | 3 | *1800* | *3* | *51.4* |
| 200r107 | 4 | 1158 | 11 | 73.2 |
| 200r108 | 3 | *1800* | *1* | – |
| 200r109 | 6 | *1801* | *1* | – |
| 200r110 | 1 | 477 | 8 | 72.1 |
| 200rc101 | 5 | 250 | 8 | 67.2 |
| 200rc102 | 7 | 1017 | 11 | 76.0 |
| 200rc103 | 12 | 43 | 6 | 59.8 |
| 200rc104 | 8 | 1007 | 10 | 72.8 |
| 200rc105 | 10 | *1801* | *1* | – |
| 200rc106 | 1 | *1801* | *1* | – |
| 200rc107 | 2 | *1801* | *1* | – |
| 200rc108 | 6 | *1800* | *1* | – |
| 200rc109 | 7 | 47 | 6 | 52.3 |
| 200rc110 | 7 | 1327 | 10 | 83.2 |

Table 4.16: Computational results for Pareto front generation for extended Solomon instances with 35 assets

| Class | Vehicles | All | |
|---|---|---|---|
| | | ST | #Opt |
| C1 | Set1 | 163 | 8/10 |
| | Set2 | 505 | 2/10 |
| R1 | Set1 | 349 | 9/10 |
| | Set2 | 280 | 2/10 |
| RC1 | Set1 | 212 | 7/10 |
| | Set2 | 113 | 3/10 |

For the first set of vehicles, only 6 out of 30 instances reached the 30-minute time limit. For the remaining 24 instances, we could generate the full Pareto fronts in less than six minutes on average. When considering the second set of vehicles, we were able to generate all the non-dominated points for only 7 out of the 30 instances.

## 4.6 CONCLUSION

The dynamic Asset Protection problem is one of the most recent variant of the Team Orienteering Problem, with time windows, synchronization constraints and two objective functions. In this section, we introduced valid inequalities and a reformulation of the mathematical model. We presented valid inequalities that rely on multiple characteristics of our problem, including bounds on deviation, minimal protection, incompatibility between assets based on time windows and resources. We also studied the structure of our problem to propose a faster mathematical formulation, that limit the scope of our model to protected assets. We managed to generate the entire Pareto front for almost all of our custom 60-asset instances in an average of 220 seconds, while the initial formulation (Merwe et al. (2017)) did not manage to obtain the extreme point with maximal deviation for any of these instances within the time limit of 1800 seconds. We also adapted the valid inequalities to apply to the mono-objective version of the problem. We managed to close all 60 benchmark 35-asset instances with 9 and 12 vehicles in 9 seconds on average, while the model without the valid inequalities (Merwe et al. (2015)) only found the optimal solution within the 600-second time limit for 27 and 47 instances respectively.

Using the reformulation and valid inequalities, the mathematical model becomes a more consistent tool to evaluate the efficiency of deployed strategies in retrospect. However, the solve time for large instances is still too significant for our model to be used in reaction to real-life situations. Multiple disruptions may occur within minutes of each other, requiring the recalculation of the Pareto front each time. It may be possible to further speed up the exact Pareto front generation, but it would probably require changes in the solution scheme. A more promising real-time approach would be to only generate a good approximation of the Pareto front, through heuristic methods. Numerous heuristic approaches for multi-objective vehicule routing problems with time windows have been studied (see for example, Garcia-Najera et al. (2011), Baños et al.

(2013), Srivastava et al. (2021), Ben-Said et al. (2022)). The reformulations and valid inequalities that we present in this section can be used within any exact or heuristic approach that relies on the mathematical formulation of the problem.

# Chapter 5

# Heuristic resolution methods

## Contents



## 5.1 Introduction

In Chapter 4, we introduced a new MIP formulation for D-APP. Although it brought improvements, it is not enough to be used to provide real-time responses to disruptions. Given the urgency of the situation, waiting for optimal solutions is not a viable option. In this chapter, we shift our focus to heuristic solution methods for D-APP. The goal is to generate high-quality approximate fronts within limited time constraints. While several efficient heuristic methods have been proposed for mono-objective APP, additional constraints in D-APP, such as deviation considerations, make it complex to adapt these methods directly.

In this chapter, we introduce two heuristic solution methods for D-APP. In Section 5.2, we present a relax-and-fix (RF) algorithm embedded in an $\epsilon$-constraint scheme. This method relies on the improvements in the MIP formulation presented in Chapter 4. In Section 5.3, we present an implementation of the Non-Dominated Sorting Genetic Algorithm (NSGA-II) for the D-APP. We introduce multiple mutation and crossover operators adapted to our problem, as well as methods to evaluate the quality of, and

repair, solutions. In Section 5.4, we analyze the computational results of both methods and highlight their respective qualities and drawbacks.

## 5.2 RELAX-AND-FIX ALGORITHM

The new formulation for D-APP introduced in Section 4.3 allowed us to generate the optimal Pareto front of larger instances than previously in the literature. However, this approach quickly hits its limits if you set a small time limit. In the following, we present a heuristic solution method that takes advantage of some characteristics of the linear relaxation of the new mathematical model.

Wolsey (2002) introduced a relaxation-based solution method called Relax-and-Fix (RF). The integritality constraints are relaxed to obtain a subproblem easier to solve, which solution is used to fix the value of some variables to generate a new easy subproblem. After several steps, a near-optimal solution of the initial model is returned. RF has been applied to many optimization problems, including lot-scheduling [Ferreira et al., 2010], lot-sizing [Toledo et al., 2015], grid-based location problems [Noor-E-Alam et al., 2012].

RF is a construction heuristic that iteratively builds a solution using the linear relaxation of the problem. At each iteration, we only optimize over a small subset of the binary variables. The remaining binary variables are either relaxed or have their value fixed to their optimal value at a previous iteration. The algorithm for the general RF approach is presented in Algorithm 2.

**Algorithm 2:** Relax-and-fix algorithm

**Require:** $\alpha$, $\beta$ : parameters
1: Construct the linear relaxation of the problem
2: Enforce the integrality constraint on $\alpha$ variables
3: Solve the model
4: **while** the solution is not integer **do**
5: Fix the value of $\beta$ binary variables to their value at last iteration
6: Enforce the integrality constraint on $\beta$ additional variables
7: Solve the model
8: **end while**

There are two parameters for the RF heuristic:

- $\alpha$: share of all variables that are integer at each iteration
- $\beta$: share of integer variables whose values are fixed at each iteration

In order to easily apply the heuristic to various sizes of instances, we express $\alpha$ as a percentage of all relaxed variables and $\beta$ as a percentage of the non-fixed integer variables.

In the following, the models are solved using a generic MIP solver. We consider a solver-specific parameter: the relative gap acceptance rate (RGA), representing the gap

between the best lower and upper bounds found by the solver. The best solution is returned before optimality is proven, as soon as the gap falls below the RGA threshold.

### 5.2.1 *Main loop*

We use the RF heuristic within the $\epsilon$-constraint scheme presented in Section 4.5.1 to replace exact resolution of the model. We cannot ensure that the solution obtained at each step is non-dominated.

**Algorithm 3:** $\epsilon$-constraint method with Relax-and-fix

```
ND = ∅
Compute bounds on deviation f_2^GLB and f_2^GUB
Let w_2 = 1 / (f_2^GUB − f_2^GLB + 1)
Let ε = f_2^GUB
while ε ⩾ 0 do
    Solve the model with objective max f_1 − w_2 f_2 and constraint f_2 ⩽ ε
    Let x* be the solution obtained with the relax-and-fix algorithm
    if x* is not dominated by a solution in S then
        S = S ∪ {x*}
    end if
    ε = f_2(x*) − 1
end while
return ND
```

The choices implied by fixing the value of certain variables may ultimately lead to infeasible solutions. In this event, and in order not to be stuck in a loop, our $\epsilon$-constraint scheme decreases the value of $\epsilon$ by one when an infeasible solution is found. However the value of $\epsilon$ is initialized to an upper bound: if the first resolution does not yield a feasible solution, it may take several iterations to sufficiently decrease $\epsilon$. We propose an initial dichomotic step to avoid being stuck in infeasible high $\epsilon$ solutions.

### 5.2.2 *Study of the relaxation*

In our model, we have three sets of decision variables: $X$, $Y$ and $Z$. In this section, we prove that if all $X$ variables have integer values, then all $Y$ and $Z$ variables also have integer value, even if the integrality constraint is relaxed on those variables. In other words, we show that we can apply the relax-and-fix algorithm on the $X$ variables only, regardless of the integrality of $Y$ or $Z$ variables.

Let suppose that $X_{ijp} \in \{0,1\}\ \forall p \in \mathcal{P}, (i,j) \in \mathcal{E}_p$. From the flow constraints (4.17) and (4.19), we can deduce that

$$\sum_{i \in \delta^-(k)} X_{ikp} \in \{0,1\}\ \forall p \in \mathcal{P}, k \in V^a \tag{5.1}$$

The expression in Equation (5.1) indicates whether vehicle $p$ visits asset $k$ or not.

INTEGRALITY OF $Y$ VARIABLES We suppose the integrality constraints on all $Y$ variables are relaxed, ie. constraint (4.26) is replaced by $0 \leqslant Y_i \leqslant 1$. In a non-dominated solution, the total protected value is maximized (Equation (4.15)): hence, the $Y$ values are maximized. We also suppose that $r_i$ has at least one non-null component, requiring at least one vehicle to meet the resource requirement of the asset (otherwise, the asset is trivially protected in any solution).

Consider an asset $k$. If no vehicle visits asset $k$, then equation (4.21) forces $Y_k = 0$. Otherwise, if $Y_k > 0$, we deduce from equation (4.21) that at least one vehicle visits asset $k$. In that case, equation (4.20) forces $Y_k = 1$. We thus showed that $Y_k \in \{0, 1\}$.

INTEGRALITY OF $Z$ VARIABLES We suppose the integrality constraints on all $Z^+$ and $Z^-$ variables are relaxed, ie. constraints (4.29) are replaced by $0 \leqslant Z^+_{ip} \leqslant 1$ and $0 \leqslant Z^-_{ip} \leqslant 1$. In a non-dominated solution, the deviation is minimized (Equation (4.16)): hence, the $Z^+$ and $Z^-$ values are minimized.

Consider a vehicle $p$ and an asset $k \in \mathcal{U}^\Phi_p$. If vehicle $p$ does not visit asset $k$, it comes from constraint (4.24) that $Z^+_{kp} = Z^-_{kp}$. The only other constraint on these variables state that they are positive: as the values are minimized, we can conclude that $Z^+_{kp} = Z^-_{kp} = 0$. Asset $k$ was not in the route of vehicle $p$ in the pre-disruption solution and is still not visited by vehicle $p$: there is no induced deviation.

If vehicle $p$ visits asset $k$, it comes from constraint (4.24) that $Z^+_{kp} = 1 + Z^-_{kp}$. As $Z^+_{kp} \leqslant 1$ and $Z^-_{kp} \geqslant 0$, the only admissible values are $Z^+_{kp} = 1$ and $Z^-_{kp} = 0$. Asset $k$ was not in the route of vehicle $p$ in the pre-disruption solution and is now visited by vehicle $p$: there is one deviation induced by its addition to the route of vehicle $p$.

Consider a vehicle $p$ and an asset $k \in \mathcal{V}^\Phi_p$. We already showed that $Y_k \in \{0, 1\}$. If asset $k$ is not protected, ie. $Y_k = 0$, then equation (4.20) states that vehicle $p$ does not visit asset $k$. Hence, constraint (4.25) becomes $Z^+_{kp} - Z^-_{kp} = 0$, implying $Z^+_{kp} = Z^-_{kp} = 0$. When an asset is not protected, it does not affect the deviation.

We now suppose asset $k$ is protected, ie. $Y_k = 1$, and vehicle $p$ visits asset $k$. It comes from constraint (4.25) that $Z^+_{kp} - Z^-_{kp} = 0$, which leads to $Z^+_{kp} = Z^-_{kp} = 0$. Asset $k$ is protected, it was in the route of vehicle $p$ in the pre-disruption solution and is still visited by vehicle $p$: there is no induced deviation.

Finally, we suppose asset $k$ is protected and vehicle $p$ does not visit asset $k$. Constraint (4.25) becomes $Z^-_{kp} = 1 + Z^+_{kp}$. As $Z^+_{kp} \geqslant 0$ and $Z^-_{kp} \leqslant 1$, the only admissible values are $Z^+_{kp} = 0$ and $Z^-_{kp} = 1$. Asset $k$ is protected, it was in the route of vehicle $p$ in the pre-disruption solution and is no longer visited by vehicle $p$: there is one deviation induced by its removal from the route of vehicle $p$.

In every scenario, we showed that $Z^+_{kp} \in \{0, 1\}$ and $Z^-_{kp} \in \{0, 1\}$.

### 5.2.3 *Variable selection strategies*

We apply the relax and fix on X variables. We can represent the X variables as a matrix $\mathcal{E} \times \mathcal{P}$. Table 5.1 shows an example with three assets, three vehicles and four arcs. All arcs are not necessarily available for all vehicles.

Table 5.1: Matrix representation of X variables

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $(x_1, x_2)$ | $X_{12p_1}$ | $X_{12p_2}$ | $X_{12p_3}$ |
| $(x_2, x_3)$ | $X_{23p_1}$ | – | $X_{23p_3}$ |
| $(x_1, x_3)$ | $X_{12p_1}$ | $X_{13p_2}$ | – |
| $(x_3, x_1)$ | $X_{31p_1}$ | $X_{31p_2}$ | $X_{31p_3}$ |

We define multiple strategies to chose the order of integrality constraints introduction on the variables X. We base our strategies on the ones defined by Toledo et al. (2015), where variables are represented in a matrix and are considered row-wise, column-wise or value-wise. We compare the strategies in Section 5.2.4.2.

MOST-FRACTIONAL This strategy is a value-wise strategy. We select the variables which values are the closest to 0.5. Figure 5.2 shows three iterations on an example with ten variables representing four arcs and three vehicles. We suppose $\alpha = 50\%$ (five variables) and $\beta = 40\%$ (two variables). We first solve the model where all X variables are relaxed: the values of the X variables are shown in the table on the left. We introduce the integrality constraints on the variables closest to 0.5: $X_{a_2p_1}$, $X_{a_1p_1}$, $X_{a_1p_2}$, $X_{a_2p_3}$ and $X_{a_4p_2}$. We then solve the model again with the newly added constraints, and obtain the results shown in the table in the middle. We select two variables to be fixed, here $X_{a_1p_1}$ and $X_{a_2p_3}$ and introduce the integrality constraint on variables $X_{a_3p_2}$ and $X_{a_4p_3}$ (shown in bold). The results of the third iteration is shown in the table on the right.

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $a_1$ | 0.4 | 0.4 | 0.9 |
| $a_2$ | 0.5 | – | 0.6 |
| $a_3$ | 0.7 | 0.8 | – |
| $a_4$ | 0.2 | 0.4 | 0.1 |

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $a_1$ | 0 | 0 | 0.9 |
| $a_2$ | 0 | – | 1 |
| $a_3$ | 0.2 | **0.6** | – |
| $a_4$ | 0.8 | 1 | **0.4** |

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $a_1$ | 0 | 0 | **0.3** |
| $a_2$ | 1 | – | 1 |
| $a_3$ | 0.1 | 1 | – |
| $a_4$ | **0.4** | 0 | 1 |

□ Relaxed variable ▒ Integer variable ■ Fixed variable

Table 5.2: First iterations of a Most-fractional strategy

ROW-WISE We chose to apply a strategy where variables are introduced and fixed row by row. In order to maintain [...], we assume that all variables linked to an arc (i.e., each row) are added at the same time. Figure 5.3 shows three iterations on an example with ten variables representing four arcs and three vehicles. We suppose $\alpha = 40\%$ (four variables) and $\beta = 50\%$ (two variables). The values of the variables when the linear relaxation is solved are given in the table on the left. We introduce the integrality

constraints on the first four variables appearing $X_{a_1p_1}$, $X_{a_1p_2}$, $X_{a_1p_3}$ and $X_{a_2p_1}$, as well as $X_{a_2p_3}$ in order to consider all variables associated with arc $a_2$. We solve the model again; the results are shown in the middle table. We fix the value of the two first variables $X_{a_1p_1}$ and $X_{a_1p_2}$, as well as $X_{a_1p_3}$ to consider all variables associated with $a_1$. We also introduce the integrality constraints on the variables of the next row $X_{a_3p_1}$ and $X_{a_3p_2}$, for a total of four free integer variables. The results of the third iteration are shown in the table on the right.

In the row-wise strategy, the order in which the arcs appear in the matrix deeply modifies the result. We propose multiple orderings that could lead to interesting results.

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $a_1$ | 0.4 | 0.3 | 0.9 |
| $a_2$ | 0.5 | – | 0.6 |
| $a_3$ | 0.7 | 0.8 | – |
| $a_4$ | 0.2 | 0.4 | 0.1 |

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $a_1$ | 0 | 1 | 0 |
| $a_2$ | 1 | – | 0 |
| $a_3$ | **0.3** | **0.3** | – |
| $a_4$ | 0.2 | 0.4 | 0.7 |

| | $p_1$ | $p_2$ | $p_3$ |
|---|---|---|---|
| $a_1$ | 0 | 1 | 0 |
| $a_2$ | 0 | – | 1 |
| $a_3$ | 1 | 0 | – |
| $a_4$ | **0.2** | **0.4** | **0.1** |

□ Relaxed variable ■ Integer variable ■ Fixed variable

Table 5.3: First iterations of an Arc-wise strategy

We associate to each asset $i$ a weight $w_i$. The assets are ordered in ascending absolute value of $w_i$. We build the ordering on the arcs based on the ordering of the assets and their *direction*. The direction of asset $i$ is determined by the sign of $w_i$: *forward* if $w_i$ is positive, or *backward* if $w_i$ is negative.

In the *forward* direction, we build the routes from the depot. If asset $i$ is ordered before asset $j$, and both assets are forward, the arcs entering asset $i$ are considered before the arcs entering asset $j$. Figures 5.1 show the first three steps for five forward assets, numbered in ascending order of $w_i$, $d$ represents the depot and $s$ the sink node. In the first iteration (Figure 5.1a), variables corresponding to arcs $(d, x_1)$ and $(d, x_2)$ are integer. In the second iteration (Figure 5.1b), arcs from the depot to $x_1$ are fixed at their values from the relaxed model, and arcs entering $x_3$ are now integer. In the third iteration (Figure 5.1c), arcs from the depot to $x_2$ are fixed, and arcs entering $x_4$ are integer.

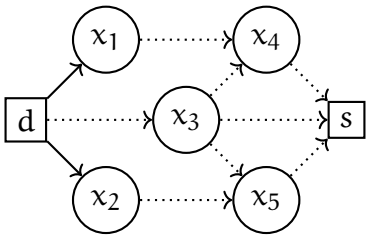
In the *backward* direction, we build the routes from the sink node. If asset $i$ is ordered before asset $j$, and both assets are backward, the arcs exiting asset $i$ are considered before the arcs exiting asset $i$.

First, we propose three orderings based on time windows. The idea is to build definitive parts of the routes.

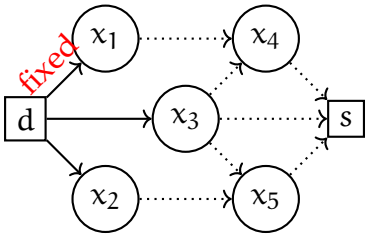
- Earliest Closing Time (ECT): $w_i = c_i$;
- Latest Closing Time (LCT) : $w_i = c_i - max(c)$;
- Half-Time Closing Time (HTCT) : $w_i = 1/(\frac{max(c)-min(c)}{2} - c_i)$.

We can note that ECT is only forward, LCT only backwards, and HTCT mixes both.

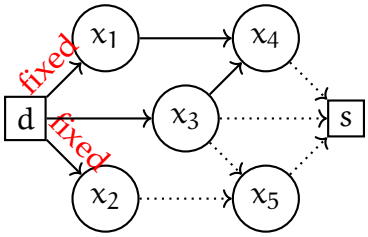
The idea behind the TW-based orderings is that when we fix a value it never leads to an infeasible solution because the arcs used before (or after) this arc are necessarily feasible (and fixed).




**(a)** First step: arcs entering assets $x_1$ and $x_2$ are integer




**(b)** Second step: arcs entering asset $x_1$ are fixed, arcs entering asset $x_3$ are integer




**(c)** Third step: arcs entering asset $x_2$ are fixed, arcs entering asset $x_4$ are integer

Relaxed arc ⟶ Free integer arc $\xrightarrow{\text{fixed}}$ Fixed arc

Figure 5.1: First iterations of a Forward strategy

Other criteria can be chosen, focusing the routing on different characteristics:

- Highest Value (HiV): $w_i = v_i$;
- Most Resources (MR): $w_i = ub_v^+(i)$;
- Random (Rdm): $w_i = rand()$.

For these criteria, we may encounter situations where the fixed variables lead to infeasible solutions.

### 5.2.4 *Preliminary results and parameter setup*

We generated five additional training instances, generated the same way as custom instances introduced in Section 4.5. We suppose that each set of parameters we are testing are almost independent.

In Section 5.2.4.1, we will study the impact of enforcing or not enforcing the integrality constraint on Y and Z variables on the solution time and how it evolves when X variables are relaxed or not. Then, we will analyse the impact of the different parameters of our algorithm: Variable selection strategies (Section 5.2.4.2), values of $\alpha$ and $\beta$ (Section 5.2.4.3) and values of RGA (Section 5.2.4.4).

#### 5.2.4.1 *Variables relaxation*

In Section 5.2.2, we demonstrated that when all X variables are integer, Y and Z variables necessarily have integer values even when relaxed.

We solved different relaxations of model (D-APP-V), without any valid inequality, on our test instances, with a time limit of 300 seconds. Average solution times by instance size are reported in Table 5.4. Each row corresponds to the enforcement of integrality constraints on X variables. The row "$0 \leqslant X \leqslant 1$" corresponds to the case where all X variables are relaxed, which is the case in the base case for the RF algorithm. The row "$X \in 0, 1$" corresponds to the case where all X variables are integer, providing insight into how the RF would react when the number of integer variables $\alpha$ is high. Each column corresponds to the enforcement of integrality constraints on variables Y and Z.

From Table 5.4, we can see that solution times are higher when Z variables are integer compared to when Z variables are relaxed, especially when X variables are also relaxed. Furthermore, relaxing the X variables when Z variables are integer leads to an increased solution time. Thus, using integer Z variables would be detrimental to the RF algorithm.

When X and Z variables are relaxed, the relaxation of Y variables have little effect. However, when Y variables are not relaxed, solution time decreases when X variables are integer. Using integer Y variables would allow the RF algorithm to be used more efficiently at each step of the solution process when integrality constraints are reintroduced, particularly with higher $\alpha$ values.

Table 5.4: Aggregated results for linear relaxation of extreme point

| | | $Y \in \{0,1\}$<br>$Z \in \{0,1\}$ | $0 \leqslant Y \leqslant 1$<br>$Z \in \{0,1\}$ | $Y \in \{0,1\}$<br>$0 \leqslant Z \leqslant 1$ | $0 \leqslant Y \leqslant 1$<br>$0 \leqslant Z \leqslant 1$ |
|---|---|---|---|---|---|
| $n = 30$ | $0 \leqslant X \leqslant 1$ | 0.29 | 34.2 | 0.11 | 0.11 |
| | $X \in \{0,1\}$ | 0.37 | 0.40 | 0.37 | 1.43 |
| 40 | $0 \leqslant X \leqslant 1$ | 33.0 | 42.0 (9/10) | 0.19 | 0.14 |
| | $X \in \{0,1\}$ | 10.1 | 13.5 | 11.0 | 24.5 |
| 50 | $0 \leqslant X \leqslant 1$ | 74.5 (8/10) | 108 (7/10) | 0.36 | 0.31 |
| | $X \in \{0,1\}$ | 55.2 (9/10) | 75.4 (8/10) | 57.1 (9/10) | 115 (7/10) |
| 60 | $0 \leqslant X \leqslant 1$ | 186 (5/10) | 206 (4/10) | 0.79 | 0.53 |
| | $X \in \{0,1\}$ | 181 (5/10) | 185 (5/10) | 192 (5/10) | 248 (3/10) |

#### 5.2.4.2 *Selecting a strategy*

In Section 5.2.3, we introduced six different criteria for row-wise variable selection. Three are based on time windows (ECT, LCT and HTCT), one is based on value (HiV), one on resources (MR) and the last one is random (Rdm).

We run the relax-and-fix algorithm with $\alpha = 0.2$ and $\beta = 0.5$ on the training instances, using two different vehicle breakdowns, with 30, 40, 50 and 60 assets. Table 5.5 shows the average hypervolume of the Pareto front obtained using each strategy.

Table 5.5: Aggregated results of RF based on variable selection strategies

| | ECT | LCT | HTCT | HiV | MR | Rdm |
|---|---|---|---|---|---|---|
| $n = 30$ | 76.27 % | 76.12 % | 75.74 % | 69.03 % | 55.33 % | 67.47 % |
| 40 | 78.55 % | 77.74 % | 78.37 % | 57.28 % | 64.75 % | 67.15 % |
| 50 | 78.68 % | 78.36 % | 78.43 % | 64.53 % | 60.08 % | 63.56 % |
| 60 | 79.37 % | 79.73 % | 76.46 % | 34.75 % | 45.43 % | 55.54 % |

We observed that the three criteria based on time windows clearly outperformed the other criteria in terms of solution quality. As noted when introduced, using HiV, MR and random criteria leads to infeasible solutions, which clearly deteriorates the quality of the fronts obtained.

There is no clear dominance between the three time window-based criteria. Further investigation is needed to detect properties that would favor one criterion over the others.

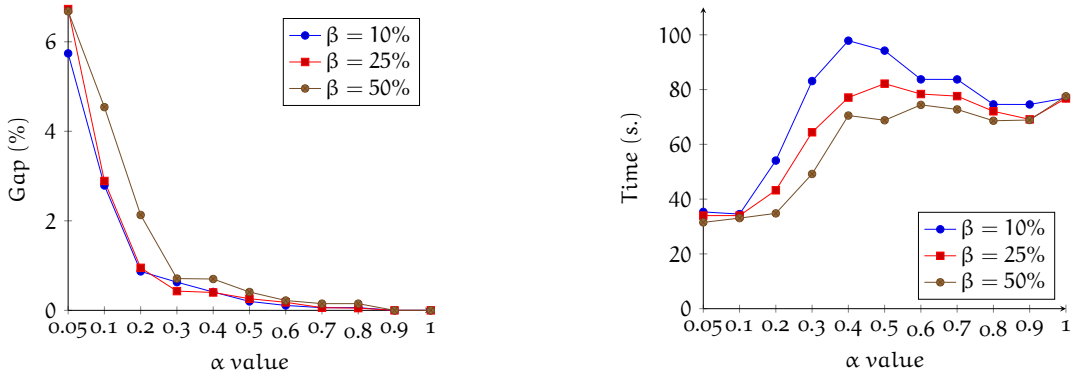
#### 5.2.4.3 *Tuning $\alpha$ and $\beta$*

In this section, we study the impact of the parameters $\alpha$ and $\beta$ on the relax-and-fix algorithm. We run the relax-and-fix algorithm on the training instances, using two different vehicle breakdowns, with 30, 40, 50 and 60 assets. The vehicle breakdowns were chosen at random, not to match the breakdowns considered in the previous section in order to avoid overfitting. We use the solver default value for RGA

(1e-4), the ECT strategy, without any valid inequalities. The tested values of $\alpha$ are $(0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)$, the tested values of $\beta$ are $(0.1, 0.25, 0.5)$. We set a time limit of 600 seconds.

Figures 5.2a and 5.2b show the evolution of the gap and solution time for instances with 50 assets. Figures 5.3a and 5.3b show the evolution of the gap and solution time for instances with 60 assets.

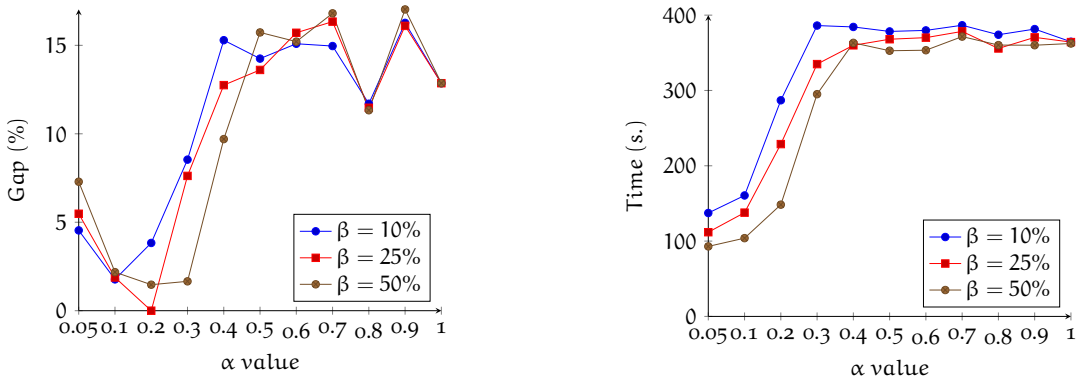
We see that for instances with 30 or 40 assets, the value of $\beta$ have little impact on the quality of the Pareto fronts returned by our algorithm. Increasing the value of $\beta$ speeds up the solution process in almost all cases, as it implies less resolutions of the relaxed model. The higher the value of $\alpha$ the closer the relaxed model is to the original model, the solutions are thus closer to the optimal solutions but are more difficult to obtain. Figure 5.2 depicts this phenomenon for instances with 50 assets. Figure 5.2a shows the quality of the Pareto front increasing quickly for small values of $\alpha$ and slowing after $\alpha = 0.2$. In parallel, Figure 5.2b shows solution time slowly increasing until $\alpha$ reaches 0.2, then sharply increasing for higher values.




**(a)** Hypervolume gap **(b)** Solution time

Figure 5.2: Computational results of $\alpha$, $\beta$ tuning for 50 assets

We can extend our analysis to instances with 60 assets, however the 600-second time limit gives us different results. The solution time follow the same pattern, as can be seen in Figure 5.3b. However, the time limit is reached for some instances with high $\alpha$ values: the Pareto front has then only a few points, if any. Figure 5.3a illustrates the drop in the quality of the Pareto front obtained when the time limit is reached, for values of $\alpha$ of 0.3 and higher.




**(a)** Hypervolume gap

**(b)** Solution time

Figure 5.3: Computational results of $\alpha$, $\beta$ tuning for 60 assets

$\alpha$ CHOICE STRATEGY We observed that the value of $\alpha$ highly influences the results obtained with the relax-and-fix algorithm. In order to balance quality and solution time, we want to generate fast points (those with low deviation) as precisely as possible while focusing on reducing computing time for high deviation points. We also want to avoid middle values (between 0.3 and 0.7), where the solution time is the highest. We discuss the choice of an $\alpha$ value, ranging from 0.2 to 0.9, based on $\epsilon$ using a sigmoid representation. The base formula for a sigmoid is $\alpha(\epsilon) = 1 - \frac{1}{1+\exp(-p_1(\epsilon-p_2))}$, where $p_1$ sets the slope of the curve and $p_2$ is the midpoint. Figure 5.4 shows an example of a sigmoid curve.

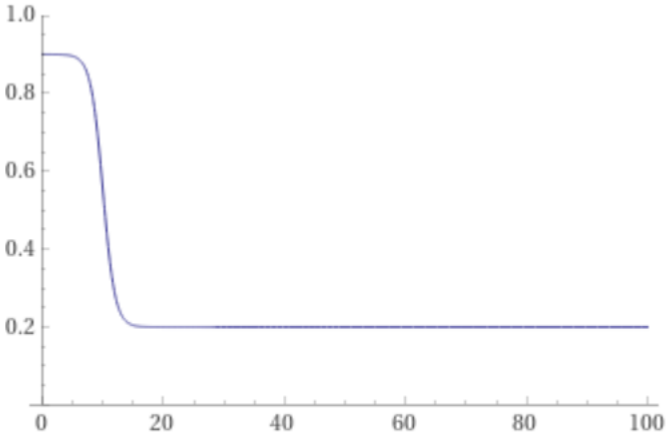
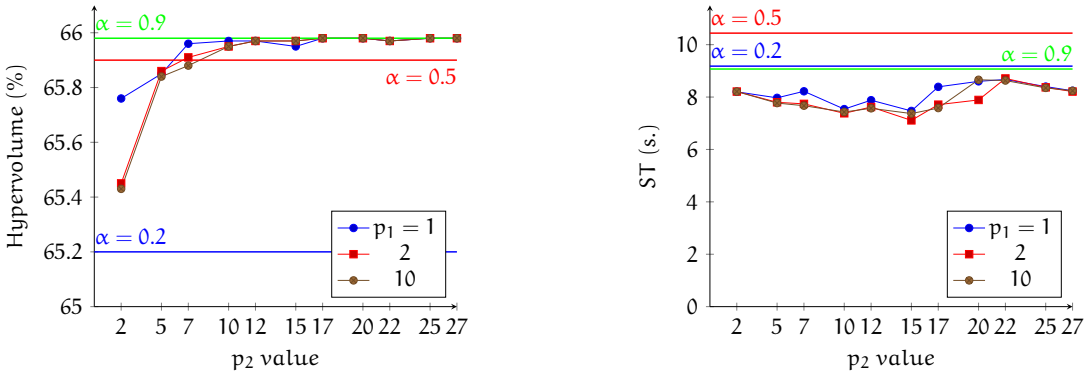
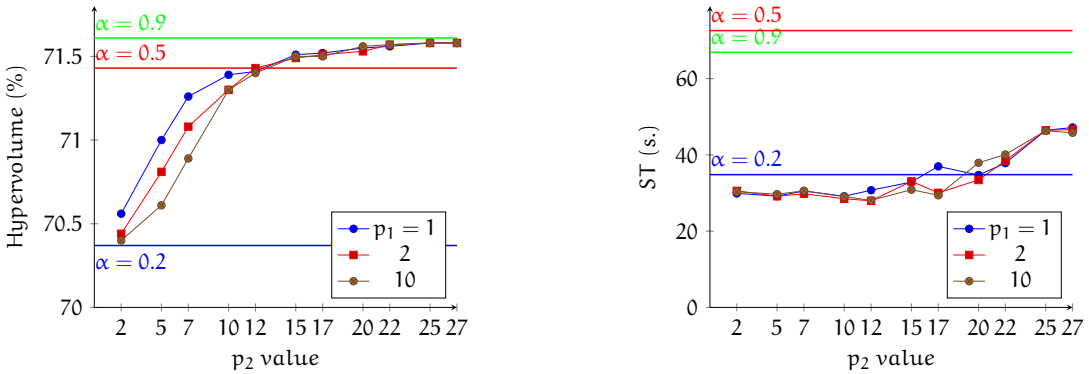



Figure 5.4: Example of a sigmoid function




**(a)** Hypervolume **(b)** Solution time

Figure 5.5: Computational results of sigmoid parameter tuning for 40 assets

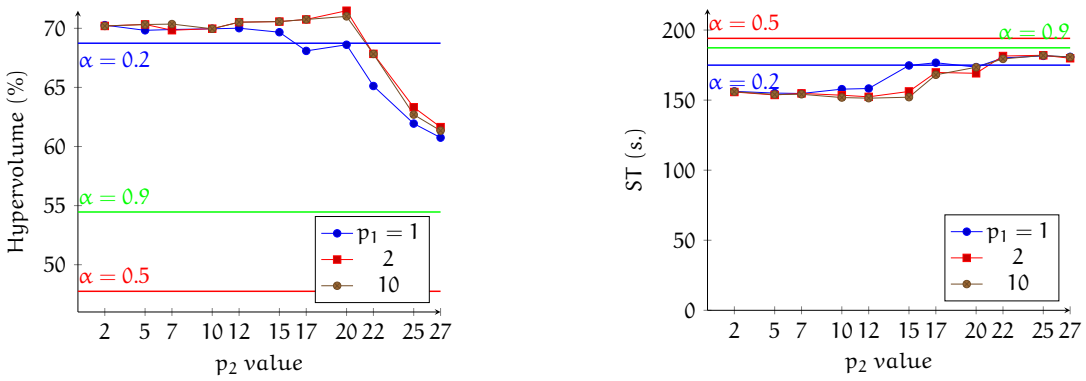



**(a)** Hypervolume **(b)** Solution time

Figure 5.6: Computational results of sigmoid parameter tuning for 50 assets

TUNING OF $p_1$ AND $p_2$ We generated the entire fronts using an $\epsilon$-constraint method with different values for the slope $p_1$ and the midpoint $p_2$, setting a time limit of 600 seconds. Figures 5.5, 5.6 and 5.7 show the average hypervolume obtained and solution time based on the values of the parameters for training instances with 40, 50 and 60 assets, respectively. We compare these results to the fronts obtained using fixed values of 0.2, 0.5, and 0.9 for $\alpha$.

Coherently with previous results, the lowest values for the midpoint $p_2$ yielded solutions of lesser quality, as the values of $\alpha$ used in the process are generally close to 0.2. However, it showed the impact of using a sigmoid, as even with a very low midpoint, we obtained a higher hypervolume than the fronts obtained with $\alpha = 0.2$, as well as a decrease in solution time. As we increased the value of $p_2$, we generated fronts closer to the optimal front, at the expense of a higher solution time. For instances with 40 assets (Figure 5.5), every value of $p_1$ and $p_2$ generated fronts faster than the three references. For instances with 50 assets (Figure 5.6), good quality fronts were generated faster than the lower quality solution obtained with $\alpha = 0.2$ for $p_2$ up to 20, and at least 20 seconds faster than the solution obtained with $\alpha = 0.9$ for all $p_2$ values.




**(a)** Hypervolume **(b)** Solution time

Figure 5.7: Computational results of sigmoid parameter tuning for 60 assets

For instances with 60 assets (Figure 5.7), we observed again the steep decrease in quality due to reaching the time limit. The use of a sigmoid function for $\alpha$ reduced the decrease in quality when the time limit is reached compared to a fixed $\alpha$ value of 0.5 or 0.9.

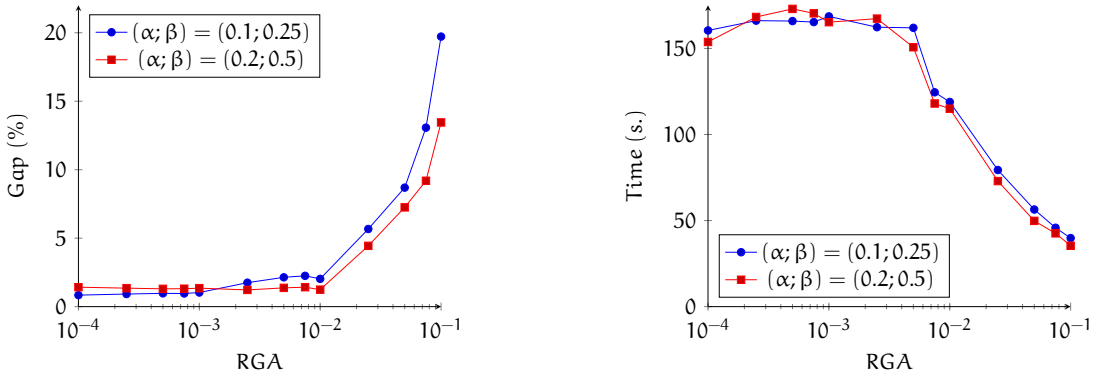
We observed that $p_1$ value only had a small impact on quality and solution time, so we decided to use $p_1 = 2$. For all instance sizes, $p_2 = 15$ represented a good compromise between quality and solution time. However, for 60 assets or more, or with a lower time limit, it may be necessary to decrease the value of $p_2$ in order to generate more points on the fronts within the time limit.

#### 5.2.4.4 *Tuning RGA*

The Relative Gap Acceptance (RGA) is a solver parameter that determines when a solution is returned based on the gap between the best integer objective and the objective of the best remaining node. In heuristic approaches, it is not necessary to wait for the solution to be proven optimal. The RGA value influences the relaxed solution obtained at each iteration.

We run the relax-and-fix algorithm on the training instances, using two different vehicle breakdowns, with 60 assets. The vehicle breakdowns were chosen at random, not to match the breakdowns considered in the previous sections in order to avoid overfitting. We used the ECT criterion, with two pairs of parameters $\alpha, \beta$: $(0.1, 0.25)$ and $(0.2, 0.5)$. We chose these values of $\alpha$ and $\beta$ because they yield solutions before the time limit for all training instances with the default RGA value. We tested values for RGA ranging from $10^{-4}$ to $10^{-1}$. We set a time limit of 300 seconds.

Figure 5.8 illustrates the results we obtained. Figure 5.8a shows the evolution of the hypervolume gap, Figure 5.8b the evolution of average solution time.




**(a)** Hypervolume gap **(b)** Solution time

Figure 5.8: Computational results of RGA tuning for 60 assets

From Figure 5.8a, we can observe that the quality of the solutions remained stable for RGA values between $10^{-4}$ and $10^{-2}$, with a gap of around 1% from the best solution. Then, the quality started deteriorating with higher RGA values, reaching two-digit gap values for values close to $10^{-1}$.

From Figure 5.8b, we can observe that the solution time started decreasing before the quality of solutions started deteriorating: an RGA value of $10^{-2}$ leads to solutions of similar quality as an RGA value of $10^{-4}$ in two-thirds of the computation time. The solution time kept rapidly decreasing as the RGA values increased.

## 5.3 NSGA-II

Several approaches extend the fast and elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) [Deb et al., 2002]. It has been efficiently applied to various multi-objective problems including but not limited to the BTOP [Mirzaei et al., 2017], the Green Vehicle Routing Problem [Jemai et al., 2012] and the Vehicle Routing Problem with Route Balancing [Jozefowiez et al., 2005].

NSGA-II is an iterative algorithm. For each generation $t$, we consider a population $R_t$ of size 2N, that is the combination of two subpopulations of size N: $P_t$, the parents, and $Q_t$, the offspring. There are three main steps in the NSGA-II algorithm, described below. The main loop of the algorithm is described in Algorithm 4. A solution $i$ has two fitness criteria relative to the current population: a rank $r_i$ and a crowding distance $d_i$. The rank represents the quality of the solution with regards to Pareto-dominance. Algorithm 5 shows the non-dominating sorting algorithm used to determine the rank of solutions within a population $P$. The crowding distance represents the quality of the solution in terms of diversification. Algorithm 6 shows how the crowding distance criterion is computed on a set of solutions $\mathcal{I}$. For more information on how these criteria are computed, we refer the reader to [Deb et al., 2002].

At generation $t$, the three steps are:

**Algorithm 4:** NSGA-II main loop

**Require:** Parent population $P_t$, Offspring population $Q_t$
**Ensure:** Parent population $P_{t+1}$, Offspring population $Q_{t+1}$

$R_t \leftarrow P_t \cup Q_t$;
$\mathcal{F} \leftarrow$ Non_dominating_sorting($R_t$);
$P_{t+1} \leftarrow \emptyset$; $i \leftarrow 1$;
**while** $|P_{t+1}| < N$ **do**
  Crowding_distance_assignment($\mathcal{F}_i$);
  $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$;
  $i \leftarrow i + 1$;
**end while**
Sort $P_{t+1}$ according to rank first, distance second;
$P_{t+1} \leftarrow P_{t+1}[1 : N]$;
$Q_{t+1} \leftarrow$ Offspring_creation($P_{t+1}$); (See Algorithm 7)
$t \leftarrow t + 1$;

**Step 1 - Initialization.** Create the population $R_t$ by combining the parent and offspring populations. Compute the rank of the solutions in $R_t$ and identify all the non-dominated fronts $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, ...)$. Compute the crowding distance of the solutions within each non-dominated front.

**Algorithm 5:** Non dominating sorting

**Require:** Population $P$
**Ensure:** Set of non-dominated fronts $\mathcal{F}$

**for** each solution $u \in P$ **do**
  Build $S_u = \{v \in P : u \succ v\}$
  Count the number $n_u$ of solutions $v \in P$ such that $v \succ u$
**end for**
$\mathcal{F}_1 \leftarrow \{u \in P, n_u = 0\}$
$i \leftarrow 1$
**while** $\mathcal{F}_i$ is not empty **do**
  **for** each solution $u \in \mathcal{F}_i$ **do**
    Decrease $n_v$ for each solution $v \in S_u$
    If $n_v$ reaches 0, add $v$ to front $\mathcal{F}_{i+1}$
  **end for**
  $i \leftarrow i + 1$
**end while**

**Step 2 - Parent population selection.** Create the parent population for next generation $P_{t+1}$ by selecting the $N$ solutions from population $R_t$. Between two solutions with different ranks, we prefer the solution with the lowest rank. If both solutions belong to the same front, we prefer the solution with the lowest crowding distance.

**Algorithm 6:** Crowding distance assignment

**Require:** Set of solutions $\mathcal{I}$
**Ensure:** Updated crowding distances for set $\mathcal{I}$
1: $l \leftarrow |\mathcal{I}|$;
2: **for** $i = 1...l$ **do**
3: $d_{\mathcal{I}[i]} \leftarrow 0$;
4: **end for**
5: **for** each objective $f_i$, $i = 1...p$ **do**
6: Sort $\mathcal{I}$ according to objective $f_i$;
7: $d_{\mathcal{I}[1]} \leftarrow \infty$;
8: $d_{\mathcal{I}[l]} \leftarrow \infty$;
9: **for** $j = 2...l-1$ **do**
10: $d_{\mathcal{I}[j]} \leftarrow d_{\mathcal{I}[j]} + f_i(\mathcal{I}[j+1]) - f_i(\mathcal{I}[j-1])$
11: **end for**
12: **end for**

**Step 3 - Offspring creation.** Create offspring population $Q_{t+1}$ from $P_{t+1}$. Details are given in Algorithm 7. The tournament operator is binary tournament, as described in [Deb et al., 2002]. Two solutions are selected at random, the solution with lowest rank is selected, or with lowest crowding distance if there is a tie. The crossover and mutation operators are discussed in Section 5.3.2. The repair and evaluation procedure is discussed in Section 5.3.3.

**Algorithm 7:** Offspring creation

**Require:** Parent population $P$, mutation rate $\mu$
**Ensure:** Offspring population $Q$
1: $Q \leftarrow \emptyset$;
2: **while** $|Q| \leqslant N$ **do**
3: $p_1 \leftarrow \mathit{tournament}(P)$;
4: $p_2 \leftarrow \mathit{tournament}(P)$;
5: $s \leftarrow \mathit{crossover}(p_1, p_2)$; (See Section 5.3.2)
6: **if** rand() $< \mu$ **then**
7: $s \leftarrow \mathit{mutate}(s)$; (See Section 5.3.2)
8: **end if**
9: $s \leftarrow \mathit{repair\_and\_evaluate}(s)$; (See Section 5.3.3)
10: $Q \leftarrow Q \cup \{s\}$;
11: **end while**
12: **return** $Q$

### 5.3.1 *Encoding*

We based the implementation of the NSGA-II algorithm for our problem on a genetic algorithm proposed for the mono-objective version of the APP with a homogeneous fleet of vehicles [Merwe, 2015].

A solution $s$ is represented by an array of integers, representing the order in which assets are visited for each vehicle. The route of a vehicle always starts at a depot and ends at the sink node. For instance, there are three vehicles in solution $[1, 2, 6, 4, 11, 1, 5, 7, 3, 11, 1, 7, 3, 11]$, the route of the first vehicle is $(1 \to 2 \to 6 \to 4 \to 11)$, the second $(1 \to 5 \to 7 \to 3 \to 11)$ and the last $(1 \to 7 \to 3 \to 11)$.

We note $\mathcal{P}_i^s$ the set of vehicles assigned to asset $i$ in solution $s$, and $\overline{\mathcal{P}_i^s}$ the set of available vehicles not assigned to asset $i$ in solution $s$.

### 5.3.2 *Operators*

First, we introduce two crossover operators: CXVAL and CXTIM. Then, we introduce two types of mutation operators. Single-change operators are mutation operators that slightly modify the solutions, by inserting or removing one asset. The multi-change operator performs a destruction/construction process on the solution.

VALID CROSSOVER OPERATOR (CXVAL). This crossover operator between two solutions $s_1$ and $s_2$ selects a vehicle at random. The route for this vehicle in $s_1$ is cut after a random asset $i_k$. The route for this vehicle in $s_2$ is also cut, after asset $j_l$. The offspring route for this vehicle is constructed by the taking the part of the route up to, and including, asset $i_k$ in $s_1$ first, and then the part of the route after asset $j_l$ in $s_2$. For example, suppose we have two routes $(i_1 \to i_2 \to \mathbf{i_3} \to i_4 \to i_5)$ and $(j_1 \to j_2 \to j_3 \to \mathbf{j_4} \to j_5 \to j_6)$, and assume the cuts happen after assets $i_3$ and $j_4$ respectively, indicated in bold. The resulting route would be $(i_1 \to i_2 \to i_3 \to j_5 \to j_6)$. The route of the second vehicle is cut in a way such that arc $(i_k, j_{l+1})$ is a valid arc. This crossover may result in duplicate assets in the route of a vehicle; we only keep the first occurrence of an asset to fix this issue.

TIME CROSSOVER OPERATOR (CXTIM). This crossover operator between two solutions $s_1$ and $s_2$ selects a time $T$ at random within the time horizon. The routes for the vehicles in $s_1$ are cut when the start time of service of the asset exceeds the chosen time, and represent the first part of the offspring routes. We then cut the routes of the vehicles in $s_2$ such that there is a valid arc between the last asset of the first part of the route and the first asset of the second part of the route. Figure 5.9 illustrates the use of the operator with two vehicles. The only arcs that may not be valid are the dotted arcs in the new solution.

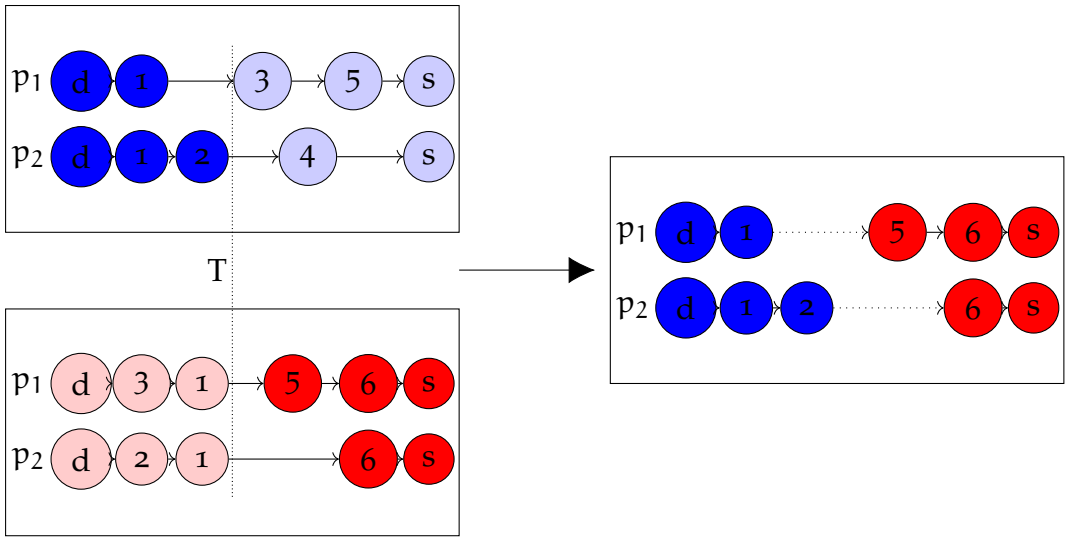



Figure 5.9: Application of the time crossover operator on solutions with two vehicles

SINGLE-CHANGE MUTATION OPERATORS. We define two different mutation operators, with same probability of being used: an insertion operator and a removal operator.

- Insertion operator. The insertion operator adds one randomly selected asset to the route of one or multiple vehicles. An asset is selected at random. The asset is added at a random valid position in the route of vehicles, taken in a random order, until the resource requirement of the asset is met.

- Removal operator. The removal operator removes one randomly selected asset from the route of one or multiple vehicles. An asset is selected at random. The asset is removed from the routes of all the vehicles it appears in.

MULTI-CHANGE MUTATION OPERATOR. We define a mutation operator that performs multiple changes on the solution, first removing multiple assets from the solution in the destruction phase, then inserting multiple protected assets in the construction phase.

During the destruction phase, the operator randomly selects $d$ assets to be removed from the current solution. The number of assets removed is randomly selected between 1 and $d_{max}$. The destruction parameter $d_{max}$ is initially set to 3. If there is no improvement on the optimal Pareto front $\mathcal{F}_1$, its value is increased, and resets to 3 when an improvement is found. In the random selection process, we can assign weights to the assets in order to favor removing assets that induce most deviation. We note $w_i^-$ the weight associated to asset $i$. The probability of selecting asset $i$ to be removed is thus $p^-(i) = w_i^- / \sum_i w_i^-$. If $w_i^- = 1$ for all assets, we have fully random behavior. Alternatively, we can use a weight based on the deviation induced by the removal of asset $i$ from solution $s$, with $\gamma$ a parameter to be determined:

$$w_i^- = \left(1 + \max(0, |\mathcal{P}_i^s| - |\overline{\mathcal{P}_i^s}|)\right)^\gamma \tag{5.2}$$

During the construction phase, the operator uses a Best Insertion Heuristic (BIH) to insert a subset of assets to the current solution. The number of assets to add is chosen randomly between $d$ and $d + c_{max}$. The construction parameter $c_{max}$ is initially set to 3. The assets to be inserted are randomly selected. We can assign weights to the assets in the selection process. We note $w_i^+$ the weight associated to asset $i$. We can use a weight based on the profit $v_i$ associated with the protection of asset $i$ and a lower bound on the deviation necessary for the protection of the asset $nb_i^+$, with $\alpha$ and $\beta$ parameters to be defined:

$$w_i^+ = v_i^\alpha / (1 + nb_i^+)^\beta \tag{5.3}$$

We want to add each asset to the route of enough vehicles for the resource protection to be met. We also want to minimize the number of vehicles we use to protect the asset. As we do not know how many vehicles will be required to meet the resource requirement, we will generate multiple insertion patterns and select the one minimizing our criterion. We detail the process in Algorithm 8. In order to account for the deviation from the pre-disruption routes, we first select the vehicles for which the asset is in the pre-disruption route. If these vehicles are not sufficient to meet the resource requirement, we continue the process with the remaining vehicles. We select the vehicles in a random order, until the protection requirement is met.

ADAPTIVE PARAMETERS. The multi-change operator relies on parameters $\alpha$, $\beta$ and $\gamma$ to control the relative importance of the different factors when associating weights to assets. They are first initialized with $\alpha = 1$, $\beta = 1$ and $\gamma = 0.5$, and then adaptively tuned during the offspring creation phase. We generate $M$ offspring solutions with slightly different values of $\alpha$, $\beta$ and $\gamma$. The values leading to the best offspring subpopulation are recorded to be used in the next iteration. All the offspring solutions generated are considered in the offspring population $Q$ of the current step.

### 5.3.3 *Repair and evaluation procedure*

A solution is represented by the route of each vehicle. It is sufficient to know the routes of the vehicles to compute the deviation from the pre-disruption routes. However, we cannot determine which assets are effectively protected: we must check if it is possible to synchronize the visits of all assigned vehicles within the time window of the asset and if the resource requirement is met by these vehicles.

Some solutions are not feasible. For instance, two vehicles may visit the same two assets in a different order, thus causing synchronization to be impossible.

Our repair procedure aims at finding the best subroutes of the solution to make it feasible and maximize the total protected value. We do not modify the order in which assets are visited by a vehicle, nor do we add new assets to the routes. The repair procedure determines which assets can actually be protected, thus contributing to the total protected value. It also provides data to correct the deviation, if unprotected assets

**Algorithm 8:** Construction: Add an asset

**Require:** Solution $S$, asset $k$, number of insertion patterns $nb\_p$
**Ensure:** A solution that protects asset $k$, if possible

**if** the available vehicles cannot meet the resource requirement **then**
  **return** $S$
**end if**
**for** $cpt = 1...nb\_p$ **do**
  $V_{cpt} \leftarrow \emptyset$; {Set of selected vehicles at iteration $cpt$}
  $cost_{cpt} = 0$;
  Determine a random order on the vehicles that prioritizes vehicles in $\mathcal{P}_k^{\Phi}$
  **for** each vehicle $p$ following the previously defined order **do**
    **if** there is a valid position in the route of vehicle $p$ **then**
      $V_{cpt} \leftarrow V_{cpt} \bigcup \{p\}$
      $cost_{cpt} \leftarrow cost_{cpt} + 1$
      **if** vehicles in $V_{cpt}$ meet the resource requirement of asset $k$ **then**
        Begin new insertion pattern (next $cpt$)
      **end if**
    **end if**
  **end for**
**end for**
Select set of vehicles $V_*$ with lowest cost
Insert asset $k$ in the routes of vehicles in $V_*$ in solution $S$
**return** $S$

have been added to the route of a vehicle, for instance. At the end of the repair procedure, we know the value of the two objective functions for the solution we have just repaired. Hence, we can use the repair procedure as the evaluation procedure for our solutions. By doing so, we also ensure that all the solutions we consider are feasible.

We propose two different MIPs used for repairing and evaluating solutions for our problem. We note $\mathcal{P}_i$ the set of vehicles that have asset $i$ in their route. We note $X_p$ the set of arcs $(i_k, i_l)$ between assets in the route of vehicle $p$, with $k < l$.

ASSET PENALIZATION. The first MIP tries to find a feasible solution from the given routes. Assets can be visited outside of their time windows, but these assets cannot be protected. Infeasibilities are lifted by removing assets entirely from the solution.

We define three sets of decision variables:

- Binary variables $Y_i$, set to 1 if asset $i$ is protected. Asset $i$ is protected when service starts within its time window and its resource requirement is met.
- Binary variables $\theta_i$, set to 1 if asset $i$ is removed from the solution.
- Continuous variables $S_i$, that represent the start time of service of asset $i$.

$$Maximize \sum_i v_i Y_i \tag{5.4}$$

$$(1 - \theta_i) \sum_p cap_p \geqslant r_i Y_i \quad \forall i \in V^a \tag{5.5}$$

$$S_i + t_{ijp} + a_i \leqslant S_j + M_1(\theta_i + \theta_j) \quad \forall p \in \mathcal{P}, (i,j) \in X_p \tag{5.6}$$

$$o_i - M_2(1 - Y_i) \leqslant S_i \leqslant c_i + M_2(1 - Y_i) \quad \forall i \in V^a \tag{5.7}$$

$$Y_i \in \{0,1\},\ \theta_i \in \{0,1\},\ S_i \in \mathbf{R} \quad \forall i \in V^a \tag{5.8}$$

Objective function (5.4) maximizes the total protected value.

Constraints (5.5) ensure that the protection requirement is met for protected assets. Assets that have been removed from the solution (with $\theta_i = 1$) cannot be protected.

Constraints (5.6) set correct start time of service for assets $i$ and $j$ when asset $i$ is visited by the vehicle before asset $j$. The order in which the assets are visited is fixed within the solution. However, as assets can be removed, we need to consider every pair of assets $(i, j)$ visited by the vehicle such that asset $i$ is visited before asset $j$.

Constraints (5.7) ensure that a protected asset is visited within its time window. Constraints (5.8) define the domain of the decision variables.

ASSIGNMENT PENALIZATION. The second MIP tries to find a feasible solution from the given routes. Infeasibilities are lifted by removing assets from the routes of individual vehicles.

We use binary variables $Y_i$ and continuous variables $S_i$. We replace variables $\theta_i$ by variables $\theta_{pi}$, set to 1 if asset $i$ is removed from the route of vehicle $p$.

$$Maximize \sum_i v_i Y_i \tag{5.9}$$

$$\sum_{p \in \mathcal{P}_i} (1 - \theta_{pi}) cap_p \geqslant r_i Y_i \quad \forall i \in V^a \tag{5.10}$$

$$S_i + t_{ijp} + a_i \leqslant S_j + M_1(\theta_{pi} + \theta_{pj}) \quad \forall p \in \mathcal{P},\ (i,j) \in X_p \tag{5.11}$$

$$Y_i + \theta_{pi} \geqslant 1 \quad \forall p,\ \forall i \in V^a \tag{5.12}$$

$$o_i - M_2(1 - Y_i) \leqslant S_i \leqslant c_i + M_2(1 - Y_i) \quad \forall i \in V^a \tag{5.13}$$

$$Y_i \in \{0,1\},\ S_i \in \mathbb{R} \quad \forall i \in V^a \tag{5.14}$$

$$\theta_{pi} \in \{0,1\} \quad \forall i \in V^a,\ p \in \mathcal{P}_i \tag{5.15}$$

Objective function (5.9) maximizes the total protected value.

Constraints (5.10) ensure that the protection requirement is met for protected assets. If asset $i$ is removed from the route of the vehicle (with $\theta_{pi} = 1$), the vehicle does not contribute to the protection.

Constraints (5.11) set correct start time of service for assets $i$ and $j$ when asset $i$ is visited by the vehicle before asset $j$, similarly to constraints (5.6).

Constraints (5.12) ensure that unprotected assets are removed from the routes of all vehicles.

Constraints (5.13) ensure that a protected asset is visited within its time window. Constraints (5.14) and (5.15) define the domain of the decision variables.

LOCAL SEARCH. After repairing a solution, we explore its neighborhood to find a dominating solution. We base our local search on the MIP used in the $\epsilon$-constraint method for the D-APP introduced in Section 4.3. We use the MIP that maximizes total protected value with deviation limited to the value of the deviation of the solution we are considering. This solution is used as a warm-start for the MIP. We set a high relative gap tolerance in our solver, meaning that the resolution will stop before the solution is proven to be optimal. For example, with a tolerance of 0.05, a solution is returned when its objective value is proven to be within 5% of the optimal value.

### 5.3.4 *Preliminary results and parameter setup*

In this section, we analyze the impact of the parameters, in order to guide the selection of efficient values. In Section 5.3.4.1, we focus on the mutation rate $\mu$. In Section 5.3.4.2, we compare the performance of our operators, evaluation models and additional components.

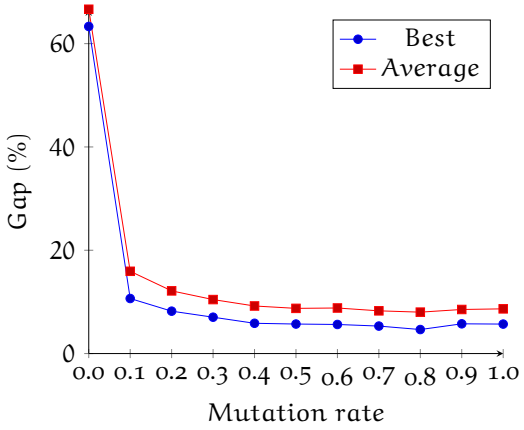



Figure 5.10: Average gap between the front obtained by NSGA-II and best known front, based on mutation rate $\mu$

#### 5.3.4.1 *Mutation rate tuning*

In this section, we want to test the influence of the mutation rate $\mu$ on the output of our algorithm. We launched the algorithm with a time limit of 60 seconds on all our benchmark instances with 30, 40, 50 and 60 assets, with two different vehicle breakdowns as the disruption.

Based on preliminary tuning work, we used fixed values for some of our parameters. The population size is set to $N = 100$. We use the time crossover operator as crossover operator and multi-change operator as mutation operator. For the choice criteria $w_i^-$ and $w_i^+$, the parameters are set to $\alpha = 1.0$, $\beta = 0.5$ and $\gamma = 1.0$. Destruction and construction parameters $c_{max}$ and $d_{max}$ are initially set to 3. The initial population is generated by applying the multi-change operator with high $c_{max}$ and $d_{max}$ values on the solution representing the initial routes.

We report in Figure 5.10 the average gap between the hypervolume of the non-dominated front $\mathcal{F}_1$ obtained with each mutation rate and the hypervolume of the best known Pareto front.

We can see that our mutation operator impacted the quality of the front we generated. We obtained the worst results when the mutation operator was disabled ($\mu = 0$), with a gap superior to 63%. The gap steeply decreased to 15% on average when the mutation operator was enabled and steadily decreased, dropping below 9% on average for $\mu = 0.5$. The gap reached its lowest point for $\mu = 0.8$ and slightly deteriorated for higer mutation rates.

#### 5.3.4.2 *Performance analysis*

In this section, we test the influence of our evaluation models and operators, and the impact of our additional components. Following preliminary work, we chose not to consider the valid crossover operator (CXVAL). Hence, we will only present results using the time crossover operator (CXTIM). We will first compare the results obtained using

our two different evaluation models, with our single-change operators and our multi-change operator. Then, we will evaluate the impact of the adaptive scheme and the local search procedure we presented.

We launched our NSGA-II algorithm five times with the different sets of operators for each of our ten benchmark instances with 30, 40, 50 and 60 assets, and two different random vehicle breakdowns as the disruption. We used the parameters presented in Section 5.3.4.1, and set the mutation rate $\mu = 0.6$.

Table 5.6 shows the results of NSGA-II within a time limit of 300 seconds. For each evaluation model (shown in row "Eval.") and operator (shown in row "Op.") combination, we give the average hypervolume of the non-dominated fronts $\mathcal{F}_1$ we obtained ($HV_{avg}$), and the hypervolume on the best run ($HV_{max}$). In the last column "$\epsilon - 300$", we indicate the hypervolume of the front obtained using the $\epsilon$-constraint method with the model introduced in Chapter 4, with a 300-second time limit. Due to the time limit, this method does not always yield the full optimal Pareto front.

Table 5.6: Comparison of the evaluation models and operators of our NSGA-II implementation

| **Eval.** | **Asset penalization** | | | | **Assignment penalization** | | | | $\epsilon - 300$ |
|---|---|---|---|---|---|---|---|---|---|
| **Op.** | **Single-change** | | **Multi-change** | | **Single-change** | | **Multi-change** | | |
| | $HV_{max}$ | $HV_{avg}$ | $HV_{max}$ | $HV_{avg}$ | $HV_{max}$ | $HV_{avg}$ | $HV_{max}$ | $HV_{avg}$ | HV |
| n=30 | 84.3% | 79.9% | 84.2% | 80.8% | 85.0% | 80.5% | 84.7% | 81.2% | 86.3% |
| 40 | 82.2% | 76.7% | 82.9% | 79.0% | 81.6% | 77.0% | 83.4% | 79.7% | 82.5% |
| 50 | 78.9% | 74.6% | 80.2% | 76.7% | 80.5% | 76.1% | 81.7% | 77.9% | 70.6% |
| 60 | 78.4% | 73.0% | 78.7% | 72.5% | 80.1% | 74.4% | 79.9% | 73.9% | 58.9% |

We can see that the second evaluation model on average gave fronts with higher hypervolume on average than the first model for the same operators. For instances with 30, 40 and 50 assets, the multi-change operator performed better than the single-change operators. The multi-change operator offers more stable results than the single-change operators, and find solutions with higher profit. For larger instances, we obtain better fronts on average than the $\epsilon - constraint$ method within the same time limit.

Based on Table 5.6, we will consider the second evaluation model with multi-change operator to evaluate our additional components. We performed a parameter analysis similar to Section 5.3.4.1 to determine good values for our adaptive method and local search parameters. We set the initial population to $N = 50$, and offspring population to $M = 50/4$. For the local search, we apply it to 5% of the solutions, with a relative gap tolerance of 0.05. Each model is run five times on each instance, to ensure the robustness of our results. In order to avoid overfitting, we selected new breakdowns at random for the benchmark instances.

Table 5.7 summarizes the results of our algorithm with no additional component, with the adaptive parameters enabled and our local search procedure. It shows the average value of the hypervolume found in the best run ($HV_{max}$) and the average value in all the runs ($HV_{avg}$).

Table 5.7: Comparison of the components of our NSGA-II implementation

| **Method** | **No component** | | **Adaptive** | | **Local Search** | | $\epsilon - 300$ |
|---|---|---|---|---|---|---|---|
| | $HV_{max}$ | $HV_{avg}$ | $HV_{max}$ | $HV_{avg}$ | $HV_{max}$ | $HV_{avg}$ | HV |
| n = 30 | 79.5% | 77.6% | 79.8% | 77.8% | 82.3% | 81.9% | 82.3% |
| 40 | 75.8% | 73.5% | 75.8% | 73.6% | 79.9% | 79.2% | 79.1% |
| 50 | 74.5% | 72.2% | 75.1% | 73.1% | 80.7% | 79.5% | 68.9% |
| 60 | 70.5% | 67.8% | 73.0% | 69.7% | 80.0% | 78.3% | 56.4% |

The adaptive component yielded similar results for instances with 30 and 40 assets and slightly better results for 50 and 60 assets when enabled. We obtained significant improvements for all instances when enabling our local search procedure, up 10% for instances with 60 assets on average. The local search procedure also improved the stability of our algorithm, reducing the gap between the best solution and the average solution for all size of instances.

## 5.4 COMPUTATIONAL RESULTS

In this section, we compare the results obtained with the RF algorithm and the NSGA-II. We solved the ten custom instances, with two different vehicle breakdowns as the deviation, with 30, 40, 50 and 60 assets. We set a time limit of 300 seconds for comparison. The NSGA-II algorithm was launched five times for each instance to test its robustness. The RF algorithm was only launched once per instance, as there is no randomness in the solution process.

Based on the tuning work done in Section 5.2.4, we run the relax-and-fix algorithm with a value of $\alpha$ following a sigmoid function with $p_1 = 2$ and $p_2 = 15$, and a value of $\beta = 0.25$. The integrality constraints was initially enforced on Y variables only. The integrality constraints on X variables was introduced using the ECT criterion. We tested two values for the RGA : $10^{-4}$ and $10^{-2}$.

Based on the tuning work done in Section 5.3.4, we run the NSGA-II algorithm with a mutation rate $\mu = 0.6$, a population size of 50, and an offspring population of 20. We used the time crossover operator (CXTIM), and the assignment penalization model for evaluation solutions. The local search was activated for 5% of the solution with a relative gap tolerance of 0.05. We tested both mutation operators: Single-change operators (SCO), and Multi-change adaptive operator (MCAO).

For reference, we also solved the instances with the exact $\epsilon-$constraint method, with a time limit of 300 seconds.

We compare the quality of the obtained fronts as detailed in Table 5.8. For NSGA-II, the best hypervolume obtained among the five runs is displayed in column "$HV_{max}$," and the average hypervolume across the five runs is shown in "$HV_{avg}$". For RF and the exact solution method (denoted as "$\epsilon - 300$), the average hypervolume is presented in the "HV" column.

Table 5.8: Computational results of heuristic solution methods on custom instances, hypervolume

| **Model Params** | **NSGA-II** | | | | **R&F** | | **$\epsilon$-300s.** |
|---|---|---|---|---|---|---|---|
| | SCO | | MCAO | | RGA $10^{-4}$ | RGA $10^{-2}$ | |
| | $HV_{max}$ | $HV_{avg}$ | $HV_{max}$ | $HV_{avg}$ | HV | HV | HV |
| n = 30 | 61.6% | 61.5% | 61.5% | 61.2% | 61.6% | 61.6% | 61.6% |
| 40 | 72.3% | 71.2% | 71.7% | 71.1% | 72.6% | 72.1% | 72.6% |
| 50 | 74.1% | 73.0% | 72.6% | 71.0% | 75.0% | 74.5% | 68.7% |
| 60 | 78.0% | 76.3% | 76.3% | 74.8% | 64.2% | 75.9% | 55.9% |

To compare solution times, we provide the "Time to Best" in Table 5.9. This time indicates the moment when a new non-dominated solution was last found. For NSGA-II, it signifies the time of the last discovery For RF and $\epsilon - 300$, it corresponds to the total computation time or the time limit if it was reached.

Table 5.9: Computational results of heuristic solution methods on custom instances, time to best (in s.)

| **Model Params** | **NSGA-II** | | **R&F** | | **$\epsilon$-300s.** |
|---|---|---|---|---|---|
| | SCO | MCAO | RGA $10^{-4}$ | RGA $10^{-2}$ | |
| n = 30 | 77.2 | 161 | 2.4 | 2.3 | 2.5 |
| 40 | 190 | 265 | 13.4 | 12.3 | 20.0 |
| 50 | 271 | 317 | 53.3 | 41.0 | 75.7 |
| 60 | 289 | 337 | 157 | 117 | 182 |

From Tables 5.8 and 5.9, we observed that:

- RF managed to generate fronts very close or equal to the optimal fronts for instances with 30 and 40 assets, all within a shorter time frame compared to the exact solution method. For these instances, NSGA-II also produced good quality fronts, with continuous improvements observed until later stages of the process.

- For instances with 50 assets, the exact solution method began reaching the time limit. Consequently, the observed hypervolume dropped as the obtained front remained partial. In contrast, RF did not reach the time limit and provided the best solutions for these instances, in less than a minute on average. NSGA-II generated fronts of slightly lower quality than RF on average but consistently outperformed the exact solution method within the time limit.

- For instances with 60 assets, RF also started reaching the 300-second time limit when using an RGA value of $10^{-4}$. he average quality of the front obtained with NSGA-II was comparable to the front obtained with RF when using an RGA value of $10^{-2}$. However, the best runs of NSGA-II outperformed RF by 2

## 5.5 CONCLUSION

In this chapter, we presented two heuristic solution methods applied to D-APP, aiming to overcome the limitations of the exact $\epsilon$-constraint method, especially when dealing with constrained computation times. First, we introduced a relax-and-fix algorithm embedded in an $\epsilon$-constraint scheme. We explored various variable selection strategies, with a particular focus on strategies based on time windows, which proved to be effective for our problem. We also examined the relaxation of the integrality constraints on the variables of the MIP formulation to maximize the utility of the new D-APP formulation.

Second, we introduced an implementation of the well-known NSGA-II algorithm. We defined multiple mutation and crossover operators tailored to our problem, as well as repair-and-evaluation procedures. Additionally, we proposed a local search procedure that uses the reformulation of D-APP.

Both methods achieved encouraging results. The RF algorithm provided nearly optimal solutions for small instances faster than the exact solution method. Moreover, it's worth noting that the RF method doesn't involve randomness, ensuring consistent results. However, it's embedded within an $\epsilon$-constraint scheme, which means that reaching the time limit returns a partial front with significantly lower quality. A trade-off between solution time and quality can be achieved by increasing the Relative Gap Acceptance parameter, as the drop in quality due to a high RGA is less significant than when the time limit is reached.

Our implementation of NSGA-II produced good approximate fronts for all instances within 300 seconds. This is a promising first step, considering the complexity of our problem, particularly in dealing with synchronization constraints and time windows. The local search procedure enabled the solution of small instances near-optimally, despite the randomness of the process. However, the evaluation of a solution is time-consuming, especially since any change to the solution necessitates a complete reevaluation. It is important to note that new non-dominated solutions were found until just before the time limit was reached. Improving the efficiency of the repair-and-evaluation process, along with more effective operators, would likely enhance the quality of the solutions.

Chapter 6

# Conclusions and future work

Wildfires are not just devastating global events; they represent multifaceted crises with far-reaching consequences. The damages they inflict on our natural environment, our cities, and most importantly, human lives, cannot be overstated. The need for effective strategies to mitigate these threats and optimize resource allocation has never been more pressing, as highlighted by the GEO-SAFE project. In this manuscript, we have aspired to contribute to this ongoing battle by focusing on the Asset Protection Problem (APP) and its disrupted counterpart, D-APP.

We started our work with an exploration of these relatively recent problems. We investigated their structure, trying to understand the complexities inherent in the protection of assets during wildfires. Our efforts yielded significant insights and improvements to their mathematical formulations. For APP and D-APP, we introduced three sets of valid inequalities, which, by design, not only accelerated the solution process but also improved its stability, particularly for smaller instances. In parallel, our reformulation for D-APP managed to generate the entire set of efficient solutions for larger instances.

Yet, we acknowledge the inherent limitations of exact solution methods, especially in the context of real-time response scenarios. Thus, we introduced two heuristic approaches for D-APP, each adapted to the specific needs and constraints of the problem. The first approach, based on our new formulation, uses a relax-and-fix strategy, while the second implements a widely adopted genetic algorithm for bi-objective problems.

## Future work

As we reflect on the findings and contributions of this thesis, we recognize that our work is a stepping stone in a larger quest to optimize resource allocation during wildfires. There are several paths for future research that hold the promise of further advancements in this field.

One such path is the development of exact solution methods. An often efficient approach in vehicle routing problems is column-generation. However, for APP, the cooperation and synchronization constraints inherent in the problem pose a unique challenge that needs to be investigated. The embedding of a column-generation technique within an $\epsilon$-constraint scheme [Glize et al., 2022] would then be possible for D-APP. Additionally, a line-generation technique, building from the work of Riera-Ledesma et al. (2021), can be of interest. In their paper, the synchronization constraints on a selective VRP are relaxed, and a cut generation component identifies resulting infeasible subsystems.

Future work may investigate the application of this method to APP, especially to account for time windows.

Another promising path, for exact solution of D-APP, where the solution times decrease as deviations become smaller, is to explore different bi-objective solution schemes, better adapted to the problem.

For the relax-and-fix approach, possible improvements include introducing randomness in the process, implementing a fix-and-optimize procedure, or embedding it in another scheme, such as a weighted sum or epsilon approach with an alternative traversal of the objective space.

In the case of NSGA-II, potential enhancements may include heuristic evaluation of solutions, bi-objective evaluation of solutions focusing on subsets of the routes, and the development of new specific operators to boost its performance.

Furthermore, a combination of methods could be explored by using a fast relax-and-fix strategy, such as one utilizing a high RGA, to generate a robust initial population for NSGA-II.

In conclusion, the ultimate goal of our research is to eventually enable real-time replanning, a critical element in responding effectively to wildfires and similar disasters. As we look to the future, our commitment to finding innovative solutions remains unwavering. By building on the foundation laid by this thesis and collaborating with fellow researchers and practitioners, we aspire to strengthen our ability to protect lives, preserve wildlife, and safeguard our cities in the face of such challenges.

# Bibliography


Amarouche, Youcef, Rym Nesrine Guibadj, Elhadja Chaalal, and Aziz Moukrim (2020). "Effective neighborhood search with optimal splitting and adaptive memory for the team orienteering problem with time windows". In: *Computers & Operations Research* 123, p. 105039 (cited on p. 20).

Aneja, Yash P and Kunhiraman PK Nair (1979). "Bicriteria transportation problem". In: *Management Science* 25.1, pp. 73–78 (cited on p. 25).

Archetti, Claudia, Dominique Feillet, Alain Hertz, and Maria Grazia Speranza (2009). "The capacitated team orienteering and profitable tour problems". In: *Journal of the Operational Research Society* 60, pp. 831–842 (cited on p. 20).

Artigues, Christian, Emmanuel Hébrard, Alain Quilliot, Peter J Stuckey, and Hélène Toussaint (2019). "MODELS AND ALGORITHMS FOR EVACUATION PLANNING FOR WILDFIRES". In: *GEO-SAFE Wildfire Conference* (cited on p. 14).

Balas, Egon (1989). "The prize collecting traveling salesman problem". In: *Networks* 19.6, pp. 621–636 (cited on p. 19).

Balinski, Michel L and Richard E Quandt (1964). "On an integer program for a delivery problem". In: *Operations research* 12.2, pp. 300–304 (cited on p. 17).

Baños, Raúl, Julio Ortega, Consolación Gil, Antonio L. Márquez, and Francisco de Toro (2013). "A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows". In: *Computers & Industrial Engineering* 65.2, pp. 286–296. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2013.01.007. URL: https://www.sciencedirect.com/science/article/pii/S0360835213000120 (cited on p. 70).

Ben-Said, Asma, Aziz Moukrim, Rym Nesrine Guibadj, and Jérôme Verny (2022). "Using decomposition-based multi-objective algorithm to solve Selective Pickup and Delivery Problems with Time Windows". In: *Computers & Operations Research* 145, p. 105867. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2022.105867. URL: https://www.sciencedirect.com/science/article/pii/S030505482200140X (cited on p. 71).

Benson, Harold P (1978). "Existence of efficient solutions for vector maximization problems". In: *Journal of Optimization Theory and Applications* 26, pp. 569–580 (cited on p. 27).

Butt, Steven E and Tom M Cavalier (1994). "A heuristic for the multiple tour maximum collection problem". In: *Computers & Operations Research* 21.1, pp. 101–111 (cited on p. 19).

Bélanger, Nicolas, Guy Desaulniers, François Soumis, and Jacques Desrosiers (2006). "Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues". In: *European Journal of Operational Research* 175.3, pp. 1754–1766. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2004.04.051 (cited on p. 42).

Chalmet, LG, L Lemonidis, and DJ Elzinga (1986). "An algorithm for the bi-criterion integer programming problem". In: *European Journal of Operational Research* 25.2, pp. 292–300 (cited on p. 25).

Clausen, Jens, Allan Larsen, Jesper Larsen, and Natalia J Rezanova (2010). "Disruption management in the airline industry—Concepts, models and methods". In: *Computers & Operations Research* 37.5, pp. 809–821 (cited on p. 21).

Croce, Federico Della, Marco Ghirardi, and Rosario Scatamacchia (2020). "An improved solution approach for the Budget constrained Fuel Treatment Scheduling problem". In: *CoRR* abs/2005.06225. arXiv: 2005.06225. URL: https://arxiv.org/abs/2005.06225 (cited on p. 13).

Cruz, MG, AL Sullivan, JS Gould, NC Sims, AJ Bannister, JJ Hollis, and RJ Hurley (2012). "Anatomy of a catastrophic wildfire: the Black Saturday Kilmore East fire in Victoria, Australia". In: *Forest Ecology and Management* 284, pp. 269–285 (cited on p. 12).

Dantzig, George B and John H Ramser (1959). "The truck dispatching problem". In: *Management science* 6.1, pp. 80–91 (cited on p. 15).

Deb, Kalyanmoy, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan (2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE transactions on evolutionary computation* 6.2, pp. 182–197 (cited on pp. 86, 88).

Defryn, Christof, Kenneth Sörensen, and Trijntje Cornelissens (2016). "The selective vehicle routing problem in a collaborative environment". In: *European Journal of Operational Research* 250.2, pp. 400–411 (cited on p. 19).

Dell'Amico, Mauro, Francesco Maffioli, and Peter Värbrand (1995). "On prize-collecting tours and the asymmetric travelling salesman problem". In: *International Transactions in Operational Research* 2.3, pp. 297–308 (cited on p. 19).

Doerner, Karl, Axel Focke, and Walter J Gutjahr (2007). "Multicriteria tour planning for mobile healthcare facilities in a developing country". In: *European Journal of Operational Research* 179.3, pp. 1078–1096 (cited on p. 19).

Dohn, Anders, Esben Kolind, and Jens Clausen (2009). "The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach". In: *Computers & Operations Research* 36.4, pp. 1145–1157. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2007.12.011 (cited on p. 42).

Drexl, Michael (2012). "Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints". In: *Transportation Science* 46.3, pp. 297–316. DOI: 10.1287/trsc.1110.0400. eprint: https://doi.org/10.1287/trsc.1110.0400 (cited on p. 42).

Eglese, Richard and Sofoclis Zambirinis (2018). "Disruption management in vehicle routing and scheduling for road freight transport: a review". In: *Top* 26, pp. 1–17 (cited on pp. 21, 42).

Ehrgott, Matthias (2005a). "Introduction". In: *Multicriteria Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–21. ISBN: 978-3-540-27659-3. DOI: 10.1007/3-540-27659-9_1. URL: https://doi.org/10.1007/3-540-27659-9_1 (cited on p. 24).

— (2005b). *Multicriteria optimization*. Vol. 491. Springer Science & Business Media (cited on p. 26).

El-Hajj, Racha, Duc-Cuong Dang, and Aziz Moukrim (2016). "Solving the team orienteering problem with cutting planes". In: *Computers & Operations Research* 74, pp. 21–30. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2016.04.008 (cited on p. 33).

Ferreira, Deisemara, Reinaldo Morabito, and Socorro Rangel (2010). "Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants". In: *Computers & Operations Research* 37.4, pp. 684–691 (cited on p. 74).

Fischetti, Matteo, Juan José Salazar González, and Paolo Toth (May 1998). "Solving the Orienteering Problem through Branch-and-Cut". In: *INFORMS Journal on Computing* 10, pp. 133–148. DOI: 10.1287/ijoc.10.2.133 (cited on p. 33).

Flores, Inmaculada, M. Teresa Ortuño, and Gregorio Tirado (2023). "A goal programming model for early evacuation of vulnerable people and relief distribution during a wildfire". In: *Safety Science* 164, p. 106117. ISSN: 0925-7535. DOI: https://doi.org/10.1016/j.ssci.2023.106117. URL: https://www.sciencedirect.com/science/article/pii/S0925753523000590 (cited on p. 14).

Garcia-Najera, Abel and John A. Bullinaria (2011). "An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows". In: *Computers & Operations Research* 38.1. Project Management and Scheduling, pp. 287–300. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2010.05.004. URL: https://www.sciencedirect.com/science/article/pii/S0305054810001176 (cited on p. 70).

Gavish, B (1984). "The delivery problem: New cutting planes procedure". In: *TIMS XXVI conference, Copenhagen*. Vol. 25 (cited on p. 16).

Geoffrion, Arthur M (1968). "Proper efficiency and the theory of vector maximization". In: *Journal of Mathematical Analysis and Applications* 22.3, pp. 618–630. ISSN: 0022-247X. DOI: https://doi.org/10.1016/0022-247X(68)90201-1. URL: https://www.sciencedirect.com/science/article/pii/0022247X68902011 (cited on p. 24).

Glize, Estèle, Nicolas Jozefowiez, and Sandra Ulrich Ngueveu (2022). "An $\varepsilon$-constraint column generation-and-enumeration algorithm for Bi-Objective Vehicle Routing Problems". In: *Computers & Operations Research* 138, p. 105570 (cited on pp. 26, 101).

Gopalakrishnan, Balaji and Ellis L Johnson (2005). "Airline crew scheduling: State-of-the-art". In: *Annals of Operations Research* 140, pp. 305–337 (cited on p. 19).

Hachemi, Nizar El, Michel Gendreau, and Louis-Martin Rousseau (Apr. 2011). "A hybrid constraint programming approach to the log-truck scheduling problem". In: *Annals of Operations Research* 184.1, pp. 163–178. DOI: 10.1007/s10479-010-0698-x (cited on p. 42).

Haimes, Yacov (1971). "On a bicriterion formulation of the problems of integrated system identification and system optimization". In: *IEEE transactions on systems, man, and cybernetics* 3, pp. 296–297 (cited on p. 26).

Hamacher, Horst W, Christian Roed Pedersen, and Stefan Ruzika (2007). "Finding representative systems for discrete bicriterion optimization problems". In: *Operations research letters* 35.3, pp. 336–344 (cited on p. 26).

Han, Hui and Eva Ponce Cueto (2015). "Waste collection vehicle routing problem: literature review". In: *PROMET-Traffic&Transportation* 27.4, pp. 345–358 (cited on p. 19).

Hansen, Pierre (1980). "Bicriterion path problems". In: *Multiple Criteria Decision Making Theory and Application: Proceedings of the Third Conference Hagen/Königswinter, West Germany, August 20–24, 1979*. Springer, pp. 109–127 (cited on p. 23).

Ioachim, Irina, Jacques Desrosiers, François Soumis, and Nicolas Bélanger (1999). "Fleet assignment and routing with schedule synchronization constraints". In: *European Journal of Operational Research* 119.1, pp. 75–90. ISSN: 0377-2217. DOI: https://doi.org/10.1016/S0377-2217(98)00343-9 (cited on p. 42).

Jemai, Jaber, Manel Zekri, and Khaled Mellouli (2012). "An NSGA-II algorithm for the green vehicle routing problem". In: *Evolutionary Computation in Combinatorial Optimization: 12th European Conference, EvoCOP 2012, Málaga, Spain, April 11-13, 2012. Proceedings 12*. Springer, pp. 37–48 (cited on p. 86).

Jozefowiez, Nicolas, Frédéric Semet, and El-Ghazali Talbi (2005). "Enhancements of NSGA II and its application to the vehicle routing problem with route balancing". In: *Artificial Evolution*. Vol. 3871. Springer, pp. 131–142 (cited on p. 86).

Laporte, Gilbert (2009). "Fifty years of vehicle routing". In: *Transportation science* 43.4, pp. 408–416 (cited on p. 15).

Laporte, Gilbert and Yves Nobert (1983). "A branch and bound algorithm for the capacitated vehicle routing problem". In: *Operations-Research-Spektrum* 5, pp. 77–85 (cited on p. 15).

Laumanns, Marco, Lothar Thiele, and Eckart Zitzler (2006). "An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method". In: *European Journal of Operational Research* 169.3, pp. 932–942 (cited on p. 26).

León, Javier, Victor M.J.J. Reijnders, John W. Hearne, Melih Ozlen, and Karin J. Reinke (Aug. 2019). "A landscape-scale optimisation model to break the hazardous fuel continuum while maintaining habitat quality". English. In: *Environmental modeling and assessment* 24, pp. 369–379. ISSN: 1420-2026. DOI: 10.1007/s10666-018-9642-2 (cited on p. 13).

Li, Jing-Quan, Denis Borenstein, and Pitu B Mirchandani (2008). "Truck schedule recovery for solid waste collection in Porto Alegre, Brazil". In: *International Transactions in Operational Research* 15.5, pp. 565–582 (cited on p. 21).

Li, Jing Quan, Pitu B. Mirchandani, and Denis Borenstein (May 2009). "Real-time vehicle rerouting problems with time windows". English (US). In: *European Journal of Operational Research* 194.3, pp. 711–727. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2007.12.037 (cited on p. 42).

Li, Yanzhi, Andrew Lim, and Brian Rodrigues (2005). "Manpower allocation with time windows and job-teaming constraints". In: *Naval Research Logistics (NRL)* 52.4, pp. 302–311. DOI: https://doi.org/10.1002/nav.20075. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20075 (cited on p. 42).

Mavrotas, George (2009). "Effective implementation of the ε-constraint method in multi-objective mathematical programming problems". In: *Applied mathematics and computation* 213.2, pp. 455–465 (cited on p. 26).

Merwe, Martijn, James Minas, Melih Ozlen, and John Hearne (Apr. 2015). "A mixed integer programming approach for asset protection during escaped wildfires". In: *Canadian Journal of Forest Research* 45. DOI: 10.1139/cjfr-2014-0239 (cited on pp. 7, 14, 29, 30, 39, 42, 55, 70).

Merwe, Martijn, Melih Ozlen, John Hearne, and James Minas (July 2017). "Dynamic rerouting of vehicles during cooperative wildfire response operations". In: *Annals of Operations Research* 254. DOI: 10.1007/s10479-017-2473-8 (cited on pp. 8, 14, 42–44, 56, 70).

Merwe, Martijn Van der (2015). "An optimisation approach for assigning resources to defensive tasks during wildfires". PhD thesis. RMIT University (cited on p. 89).

Mirzaei, Milad Haj, Koorush Ziarati, and Mohammad-Taghi Naghibi (2017). "Bi-objective version of team orienteering problem (BTOP)". In: *2017 7th international conference on computer and knowledge engineering (ICCKE)*. IEEE, pp. 1–7 (cited on p. 86).

Mu, Qianxin, Zhuo Fu, Jens Lysgaard, and Richard Eglese (Apr. 2011). "Disruption management of the vehicle routing problem with vehicle breakdown". In: *JORS* 62, pp. 742–749. DOI: 10.1057/jors.2010.19 (cited on p. 42).

Noor-E-Alam, Md and John Doucette (2012). "Relax-and-fix decomposition technique for solving large scale grid-based location problems". In: *Computers & Industrial Engineering* 63.4, pp. 1062–1073 (cited on p. 74).

Nuraiman, Dian, Melih Ozlen, and John Hearne (2020). "A spatial decomposition based math-heuristic approach to the asset protection problem". In: *Operations Research Perspectives* 7, p. 100141. ISSN: 2214-7160. DOI: https://doi.org/10.1016/j.orp.2020.100141 (cited on pp. 29, 39).

Ozdamar, Linet, Dilek Tüzün, Elifcan Yaşa, and Biket Ergüneş (Jan. 2017). "Disaster relief routing in limited capacity road networks with heterogeneous flows". In: *Journal of Industrial & Management Optimization* 13, pp. 1–14. DOI: 10.3934/jimo.2018011 (cited on p. 14).

Ozlen, Melih and Meral Azizoglu (Nov. 2009). "Multi-objective integer programming: A general approach for generating all non-dominated solutions". In: *European Journal of Operational Research* 199, pp. 25–35. DOI: 10.1016/j.ejor.2008.10.023 (cited on pp. 26, 55).

Pausas, Juli G, Joan Llovet, Anselm Rodrigo, and Ramon Vallejo (2009). "Are wildfires a disaster in the Mediterranean basin?–A review". In: *International Journal of wildland fire* 17.6, pp. 713–723 (cited on p. 12).

Peña, Quentin, Aziz Moukrim, and Mehdi Serairi (2022). "An elitist non-dominated heuristic resolution for the dynamic asset protection problem". In: *15th International Conference on Artificial Evolution* (cited on p. 8).

Perboli, Guido, Roberto Tadei, and Edoardo Fadda (Feb. 2018). "New Valid Inequalities for the Two-Echelon Capacitated Vehicle Routing Problem". In: *Electronic Notes in Discrete Mathematics* 64, pp. 75–84. DOI: 10.1016/j.endm.2018.01.009 (cited on p. 33).

Peña, Quentin, Mehdi Serairi, and Aziz Moukrim (2023). "Reformulation and valid inequalities for the disrupted asset protection problem during an escaped wildfire". submitted for publication (cited on p. 8).

Riera-Ledesma, Jorge and Juan-José Salazar-González (2021). "Selective routing problem with synchronization". In: *Computers & Operations Research* 135, p. 105465 (cited on p. 101).

Robinson, Julia (1949). *On the Hamiltonian game (a traveling salesman problem)*. Rand Corporation (cited on p. 15).

Roozbeh, Iman, Melih Ozlen, and John W. Hearne (2018). "An Adaptive Large Neighbourhood Search for asset protection during escaped wildfires". In: *Computers & Operations Research* 97, pp. 125–134. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2018.05.002 (cited on pp. 29, 38, 39).

Running, Steven W (2006). "Is global warming causing more, larger wildfires?" In: *Science* (cited on p. 12).

Short, KC (2014). "A spatial database of wildfires in the United States, 1992-2011". In: *Earth System Science Data* 6.1, pp. 1–27 (cited on p. 12).

Srivastava, Gaurav, Alok Singh, and Rammohan Mallipeddi (2021). "NSGA-II with objective-specific variation operators for multiobjective vehicle routing problem with time windows". In: *Expert Systems with Applications* 176, p. 114779. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2021.114779. URL: https://www.sciencedirect.com/science/article/pii/S0957417421002207 (cited on p. 71).

Steuer, Ralph E and Eng-Ung Choo (1983). "An interactive weighted Tchebycheff procedure for multiple objective programming". In: *Mathematical programming* 26, pp. 326–344 (cited on p. 25).

Sylva, John and Alejandro Crema (2004). "A method for finding the set of non-dominated vectors for multiple objective integer linear programs". In: *European Journal of Operational Research* 158.1, pp. 46–55 (cited on p. 25).

Toledo, Claudio, Márcio Arantes, Marceloyukiobressan Hossomi, Paulo França, and Kerem Akartunalı (Oct. 2015). "A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems". In: *Journal of Heuristics* 21, pp. 1–31. DOI: 10.1007/s10732-015-9295-0 (cited on pp. 74, 77).

Tsiligirides, Theodore (1984). "Heuristic methods applied to orienteering". In: *Journal of the Operational Research Society* 35, pp. 797–809 (cited on p. 19).

Ulungu, Ekunda Lukata and Jacques Teghem (1995). "The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems". In: *Foundations of computing and decision sciences* 20.2, pp. 149–165 (cited on p. 26).

Veeraswamy, Anand, Edwin R Galea, Lazaros Filippidis, Peter J Lawrence, Simo Haasanen, Robert J Gazzard, and Thomas EL Smith (2018). "The simulation of urban-scale

evacuation scenarios with application to the Swinley forest fire". In: *Safety science* 102, pp. 178–193 (cited on p. 14).

Wang, Xuping, Xu Wu, and Xiangpei Hu (2010). "A study of urgency vehicle routing disruption management problem". In: *2010 WASE International Conference on Information Engineering*. Vol. 3. IEEE, pp. 452–455 (cited on p. 21).

Wolsey, Laurence A (2002). "Solving multi-item lot-sizing problems with an MIP solver using classification and reformulation". In: *Management Science* 48.12, pp. 1587–1602 (cited on p. 74).

Yahiaoui, Ala-Eddine, Aziz Moukrim, and Mehdi Serairi (2022). "GRASP-ILS and set cover hybrid heuristic for the synchronized team orienteering problem with time windows". In: *International Transactions in Operational Research* n/a.n/a. DOI: https://doi.org/10.1111/itor.13111. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/itor.13111. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.13111 (cited on pp. 29, 39).

Yu, Gang and Xiangtong Qi (2004). *Disruption management: framework, models and applications*. World Scientific (cited on p. 21).

Zitzler, Eckart and Lothar Thiele (1998). "Multiobjective optimization using evolutionary algorithms—a comparative case study". In: *International conference on parallel problem solving from nature*. Springer, pp. 292–301 (cited on p. 27).

# List of Abbreviations

| | |
|---|---|
| **VRP** | **Vehicle Routing Problem** |
| **TOP** | **Team Orienteering Problem** |
| **APP** | **Asset Protection Problem** |
| **D-APP** | **Disrupted Asset Protection Problem** |
| **MIP** | **Mixed Integer Programming** |
| **RF** | **Relax-and-Fix** |
| **NSGA** | **Non-dominated Sorting Genetic Algorithm** |
| **HV** | **Hypervolume** |
| **PF** | **Pareto Front** |