Les Cahiers du GERAD ISSN: 0711–2440

Towards resilience: Primal large-scale re-optimization

E. M. Er Raqabi, Y. Wu, I. El Hallaoui, F. Soumis

G–2023–28

August 2023

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2023
– Bibliothèque et Archives Canada, 2023

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2023
– Library and Archives Canada, 2023

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Towards resilience: Primal large-scale re-optimization

El Mehdi Er Raqabi [a, b]

Yong Wu [c]

Issmaïl El Hallaoui [a, b]

François Soumis [a, b]

[a] *Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal (Qc), Canada, H3T 3A7*

[b] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[c] *Department of Business Strategy and Innovation, Griffith University, Gold Coast Campus, QLD 4222 Australia*

el-mehdi.er-raqabi@polymtl.ca

August 2023
Les Cahiers du GERAD
G–2023–28

Abstract : Perturbations are universal in supply chains, and their appearance is getting more frequent in the past few years. These perturbations affect industries and could significantly impact production, quality, cost/profitability, and consumer satisfaction. In large-scale contexts, companies rely on mathematical optimization. Still, these companies must remain resilient to perturbations. In such a case, re-optimization can support companies in achieving resilience by enabling them to adapt to changing circumstances and challenges in real-time. In this paper, we design a generic and scalable resilience re-optimization framework. We model perturbations, recovery decisions, and the resulting re-optimization problem to maximize resilience. We leverage the primal information through fixing, warm-start, valid inequalities, and machine learning. We conduct extensive computational experiments on a real-world large-scale problem highlighting that local optimization is enough to recover after perturbations and demonstrating the power of our proposed framework and solution methodology.

Keywords : Large-scale optimization, re-optimization, resilience, primal information, machine learning, perturbation

---

Acknowledgements: This work was generously supported by the *OCP Group*, the *Fonds de Recherche du Québec – Nature et Technologies* (FRQNT) through a Ph.D. Excellence Scholarship, the *Institute for Data Valorisation* (IVADO) through another Ph.D. Excellence Scholarship, and the *Group for Research in Decision Analysis* (GERAD) through the provision of computing facilities for conducting experiments.

1 Introduction

Perturbations are universal in supply chains (SCs), and their appearance has been more frequent in the past few years. These perturbations affect industries and organizations and could significantly impact production, quality, cost/profitability, and consumer satisfaction. They can be caused by several factors, including global events, localized incidents with global impact, and shifting environmental conditions. Global events such as the COVID-19 pandemic (Chakraborty and Maity, 2020; Shen et al., 2020), the Ukraine war (Mbah and Wasum, 2022), and the food crisis (Gliessman, 2022) have brought about unprecedented changes, creating uncertainty about what the future will look like. Localized incidents, such as the blockage of the Suez Canal by Ever Given (Lee and Wong, 2021), can create a huge challenge to global logistics. Furthermore, environmental conditions are shifting as we become more aware of issues such as climate change (Winn et al., 2011) and natural disasters (Akkermans and Van Wassenhove, 2018), which in turn have led to changes in operations and SC management, consumer behavior, and government policies, etc.

The perturbations highlighted above increase the operations management complexity within and among corporations. This increased complexity generates large and complex optimization problems. Given these problems' size, manual solving is intractable. Thus, companies invest heavily in mathematical optimization tools. Still, these large-scale optimization problems involve combinatorial mathematical models with complex multiobjective functions and millions of constraints and variables, making their solving costly. In some, no feasible solutions can be rapidly identified, even using off-the-shelf optimizers. In such a case, organizations rely on sophisticated operations research (OR) techniques to generate (near)-optimal solutions, or even feasible solutions, in an *acceptable* amount of time, which may still be relatively *long* if run repetitively.

While using mathematical optimization for large-scale optimization problems, companies must remain resilient to perturbations. Among many definitions (Barroso et al., 2015), resilience can be defined as the ability of a system (e.g., company, organization, SC) to return to its original state or move to a new, more desirable state after being disturbed (Christopher and Peck, 2004). To do so, organizations should stay informed and adapt to any changes to sustain their operations and performance in the market. In several contexts, recovering after being perturbed and adapting to changes must be quick. Thus, companies cannot afford to optimize after each change using off-the-shelf optimizers or sophisticated OR techniques because recovery time might be relatively *long*.

Re-optimization can support companies in achieving resilience by enabling them to adapt to changing circumstances and challenges in real-time. It is an effective and efficient way to recover the original state quickly or move to a better one. Compared to optimizing from scratch after each change, re-optimizing from a previous state leverages the existing *primal* information, significantly reducing the recovery time, i.e., solving the updated and refined optimization models that reflect the new data and changing circumstances. Such gain can allow companies to re-optimize several times, i.e., whenever a perturbation affects its system. Furthermore, re-optimization also supports companies in identifying and mitigating risks before happening. By continuously analyzing data and considering potential trends, these companies can proactively identify and address vulnerabilities in their operations, increasing further resilience.

The present article has a fivefold contribution: (1) We design a generic and scalable resilience re-optimization framework; (2) We identify and model perturbations, identify and model recovery decisions, and highlight the need for resilience; (3) We model the re-optimization problem to maximize resilience and solve it using a variant of the large neighborhood search (LNS) metaheuristic; (4) We leverage the primal information using fixing, warm-start, valid inequalities, and machine learning (ML) techniques; and (5) We conduct extensive computational experiments on a real-world large-scale optimization problem, which highlight that local optimization is enough to recover quickly after perturbations.

To illustrate our research, we consider a real-world large-scale optimization problem for which perturbations are related to the maritime distribution side and happen on a port. The goal is to quickly recover after perturbations and reach an optimal solution while being as *satisfactory* as possible and remaining as *close* as possible to the perturbed solution, which was previously the optimal solution.

We organize the rest of the paper as follows: We first present an overview of the relevant literature in Section 2. Then, we highlight the generic re-optimization framework in Section 3. Section 4 is devoted to a detailed description of the considered problem with its formulation. Section 5 presents the solution methodology. We highlight the experimental design in Section 6, show the computational results and managerial insights in Section 7, and conclude the paper in Section 8.

2 Literature review

In this section, we present the relevant resilience and re-optimization literature before positioning our research.

2.1 Resilience

Supply chain resilience (SCR) has been studied from both qualitative and quantitative perspectives. The former, which dominated in the past (Kamalahmadi and Parast, 2016), consists of approaching SCR in a rather qualitative manner, providing a set of strategies that can increase SCR without providing performance metrics to quantify the impact of a particular strategy on SC operations. The latter, which is more dominant in recent years (Hosseini et al., 2019a), consists of mathematically and analytically modeling and measuring SCR.

From a qualitative perspective, Yang et al. (2009) study a manufacturer that faces a supplier privileged with private information about supply perturbations. They investigate how the risk-management strategies of the manufacturer change and examine whether risk-management tools are more or less valuable in the presence of such asymmetric information. Yang and Fan (2016) compares the disruption mitigation effects of three information management strategies using control theory modeling and simulation. They show that SCs with popular information management strategies are not evidently more stable than traditional ones. Chopra et al. (2021) offer the notion of "commons" at different levels (company, private across the company, and government-sponsored across-industry sectors) and discuss how the creation of such commons enabled firms to be both efficient during normal times and resilient against the disruptions resulting from COVID-19.

From a quantitative perspective, Chen and Miller-Hooks (2012) design an indicator for network resilience that quantifies the ability of an intermodal freight transport network to recover from disruptions due to natural or human-caused disasters. The indicator considers the network's inherent ability to cope with the negative consequences of disruptions as a result of its topological and operational attributes. They propose a stochastic mixed-integer program (SMIP) for quantifying network resilience and identifying an optimal post-event course of action (i.e., set of activities) to take. They solve it using a technique that combines concepts from Benders decomposition (BD), column generation (CG), and Monte Carlo (MC) simulation. An et al. (2015) present a scenario-based stochastic mixed-integer non-linear program (SMINLP) model that integrates facility disruption risks, en-route traffic congestion, and in-facility queuing delay into an integrated facility location problem. After deriving lower and upper bounds, they tackle it using Lagrangian relaxation (LR). Khaled et al. (2015) propose a mixed-integer programming (MIP) model for making up and routing trains in a disruptive situation to minimize the system-wide total cost, including classification time at yards and travel time along links. They solve it using an iterative heuristic algorithm. Khalili et al. (2017) present a two-stage scenario-based mixed stochastic-possibilistic programming (TSMSP) model for the integrated production and distribution planning problem in a two-echelon supply chain over a midterm horizon under risk. They solve it via a multi-step approach. Sahebjamnia et al. (2018) propose an integrated business continuity and disaster recovery planning (IBCDRP) model to build organizational resilience that can respond to multiple disruptive incidents, which may occur simultaneously or sequentially. A multi-objective mixed-integer robust possibilistic programming (MIRPP) model, which accounts for sensitivity and feasibility robustness, is formulated. They tackle it using a two-phase approach. Hosseini et al. (2019b) provide a stochastic bi-objective mixed integer programming model to support the decision-making in how and when to use both proactive and reactive strategies in supplier selection and order allocation. They solve it using a two-step approach. Elluru et al. (2019) stipulate that the supply chain distribution network broadly comprises two major decisions: facility location and vehicle routing. Then, they address these distribution decisions jointly as a location-routing problem and solve it using the solver LINGO. Sawik (2019) proposes a two-period modeling approach for supply chain disruption mitigation and recovery and compares it with a multi-period approach. The models are SMIP with no solution methodology. We summarize the relevant resilience literature in Table 1.

Table 1: Summary of relevant resilience literature

Author (Year)	Approach	Context	Model	Algorithm	Objective
Yang et al. (2009)	Qualitative	Manufacturing	—	—	—
Chen and Miller-Hooks (2012)	Quantitative	Transportation	SMIP	BD+CG+MC	Max Demand
An et al. (2015)	Quantitative	Location	SMINLP	LR	Min Cost
Khaled et al. (2015)	Quantitative	Transportation	MIP	Heuristic	Min Cost
Yang and Fan (2016)	Qualitative	SC	—	—	—
Khalili et al. (2017)	Quantitative	SC	TSMSP	Multi-step	Min Cost
Sahebjamnia et al. (2018)	Quantitative	Manufacturing	MIRPP	Two-phase	Min Loss
Elluru et al. (2019)	Quantitative	SC	SMIP	LINGO	Min Cost
Hosseini et al. (2019b)	Quantitative	SC	SMIP	Two-step	Max Distance
Sawik (2019)	Quantitative	SC	SMIP	—	Min Cost
Chopra et al. (2021)	Qualitative	Industry	—	—	—
This Paper	Quantitative	SC	MILP	Heuristic	Max Resilience

2.2 Re-optimization

Re-optimization is an efficient way to ensure resilience in large-scale contexts. We distinguish two types of re-optimizations: major and minor. Major re-optimizations happen after a disruption and are more strategic/tactical and less frequent (e.g., once, annually). They are often conducted from scratch and usually require an exact algorithm. On the contrary, minor re-optimizations occur after a perturbation and are more tactical/operational and frequent (e.g., weekly, daily, real-time). They are conducted more from a previous solution than from scratch and usually require a heuristic algorithm.

From a major perspective, Bruno et al. (2021) rely on re-optimization to re-engineer a system. The latter consists of reorganizing the collection system of an Italian postal service provider. They model the problem as a MIP that identifies the number of postboxes (currently located in an urban area) to be closed. They solve it using a two-phase methodology based on mathematical programming. Chen and Miller-Hooks (2012), cited in Section 2.1, also rely on major re-optimization. We recall that they propose a SMIP for quantifying network resilience and identifying an optimal post-event course of action (i.e., set of activities).

From a minor perspective, D'Ariano et al. (2010) present a graph formulation for the train running profile problem. A conflict solution system is developed, that models the train scheduling problem as an alternative graph. From a network point of view, the optimal solution can be improved by modifying the speed profiles locally for the individual train routes. A constructive heuristic algorithm for the dynamic modification of running times during operations is proposed that satisfies the timetable constraints of train orders and routes and guarantees the feasibility of the running profile while considering the properties of the signaling and train protection systems in use. Archetti et al. (2013) consider the re-optimization of the Rural Postman Problem (RPP) given an instance and its optimal solution. They study the problem of finding a satisfactory feasible solution after a perturbation (new edge

Table 2: Summary of relevant re-optimization literature

Author (Year)	Type	Context	Model	Algorithm	Objective
D'Ariano et al. (2010)	Minor	Transportation	Graph	Heuristic	Max Trajectory
Chen and Miller-Hooks (2012)	Major	Transportation	SMIP	Exact	Max Demand
Archetti et al. (2013)	Minor	Transportation	—	Heuristic	Min Cost
Dong et al. (2018)	Minor	Transportation	MIP	CPLEX	Min Cost
Schieber et al. (2018)	Minor	CRO	MIP	Heuristic	Min Distance
Doerr et al. (2019)	Minor	CRO	MO	Heuristic	Min Distance
Hassani et al. (2020)	Minor	Personnel	MIP	Heuristic	Min Cost
Bruno et al. (2021)	Major	Location	MIP	Exact	Min Distance
Hasani et al. (2021)	Minor	Personnel	MIP	Labeling	Min Cost+Distance
This Paper	Minor	SC	MIP	Heuristic	Max Resilience

added or removed) of the instance has occurred and tackle it heuristically. Schieber et al. (2018) develop a general framework for combinatorial re-optimization (CRO), encompassing classical objective functions as well as the goal of minimizing the transition cost from one solution to the other. Using their model, they derive re-optimization and re-approximation algorithms for several combinatorial re-optimization problem classes. Dong et al. (2018) study a maritime inventory routing MIP problem in which shipments between production and consumption nodes are carried out by a fleet of vessels. In the face of new information and uncertainty, this optimization model has to be resolved as the horizon is rolled forward. They discuss how to account for different sources of uncertainty and present a rolling-horizon re-optimization framework that allows studying different policies that impact the quality of the implemented solution. They use the solver CPLEX for re-optimization. Doerr et al. (2019) show that evolutionary algorithms can have unexpected difficulties to solve re-optimization problems, which build on a previously good feasible solution. Then, they propose a simple diversity mechanism that works for various mathematical optimization (MO) problems, including the LeadingOnes, linear functions with modified uniform constraints, and the minimum spanning tree problems. Hassani et al. (2020) develop a fast re-scheduling heuristic that can be used to solve the personnel re-scheduling problem in a context where the employees can be assigned to a wide variety of shifts such as in the retail industry. Hassani et al. (2021) propose a fast re-scheduling heuristic that can be used to correct minor disruptions in a retail industry context where employees can be assigned to a wide variety of shifts, starting and ending at numerous times. This heuristic can compute a set of approximate Pareto-optimal solutions that achieve a good compromise between cost and number of shift changes. It can be seen as a labeling algorithm that partially explores the network defined by the edges of the convex hull of the solutions of an integer program. We summarize the relevant re-optimization literature in Table 2.

2.3 Our research

In our paper, we seek SCR via re-optimization. As far as we acknowledge, the only close research paper to ours is the one in the intersection of Sections 2.1 and 2.2, which is the work of Chen and Miller-Hooks (2012). While this paper and ours are both quantitative, the former belongs to the major re-optimization (disruption) case, while ours belongs to the minor re-optimization (perturbation) case. Furthermore, we quickly re-optimize from a previous solution, while Chen and Miller-Hooks (2012) re-optimize from scratch. All other papers belong to either Section 2.1 or Section 2.2 but not both.

Within the resilience and re-optimization pieces of literature, we distinguish our paper as follows. First, we propose a generic and scalable resilience/re-optimization framework. Second, using the distribution side of a global SC (large-scale problem) case to illustrate, we quantify resilience and build a new MIP model from the model in Er Raqabi et al. (2023b), where some constraints and variables are kept, and resilience is the objective function. Third, we leverage the primal information using ML, valid inequalities, warm-starting, and fixing techniques to reach satisfactory solutions quickly to suit the need for SCR via re-optimization.

3 Resilience/re-optimization framework

In this section, we introduce the generic and scalable Resilience/Re-Optimization (RRO) framework. We then offer discussions from a managerial perspective based on the supply chain operations reference (SCOR) model.

3.1 Framework

Let us consider a large-scale optimization problem (*Original Problem*) modeling the SC of a company. Using solvers and OR techniques, the company obtains an optimal solution. Faced with uncertainties (*Perturbation(s)*), this solution may no longer be feasible. In such a case, we note that solution $\cancel{q}^*$. Thus, the company wants to be resilient and to quickly re-optimize and reach a near-optimal, if not optimal, solution q^* as *satisfactory* as possible as solution $\cancel{q}^*$.

To remain resilient, the company has to define (*Resilience Definition*) and model (*Resilience Modeling*) resilience. The *resilience definition* must be clear to allow the company model accordingly. Furthermore, the company has to identify and model *perturbation(s)*. Then, it has to identify and model the set of actions (*Decision(s)*) to take. Following these aspects, it can formulate the *re-optimization problem* to maximize resilience while taking into consideration the *original problem*. We refer to this first stage (red frame) as the *problem definition* stage.

After formulating the *re-optimization problem*, the company can design its *re-optimization approach* while leveraging the *primal information* using the *solution* $\cancel{q}^*$, the *original problem*, and the company's *history* (e.g., solutions history, problem knowledge, accumulated expertise). Such information is relevant since we do not want to optimize from scratch. The qualification *primal* is borrowed from the optimization lexicon and is used mainly to distinguish between dual and primal methods. The former does not take into consideration the accumulated information in the optimization process, while the latter leverages the accumulated information to reach optimality quickly. The *re-optimization approach* allows reaching a feasible solution q^* as *close* as possible to the no-longer feasible solution $\cancel{q}^*$ and hence causes the least amount of changes in response to the *perturbations*. We refer to the second stage (green frame) as the *solution methodology* stage. We reflect both stages in the RRO framework in Figure 1.

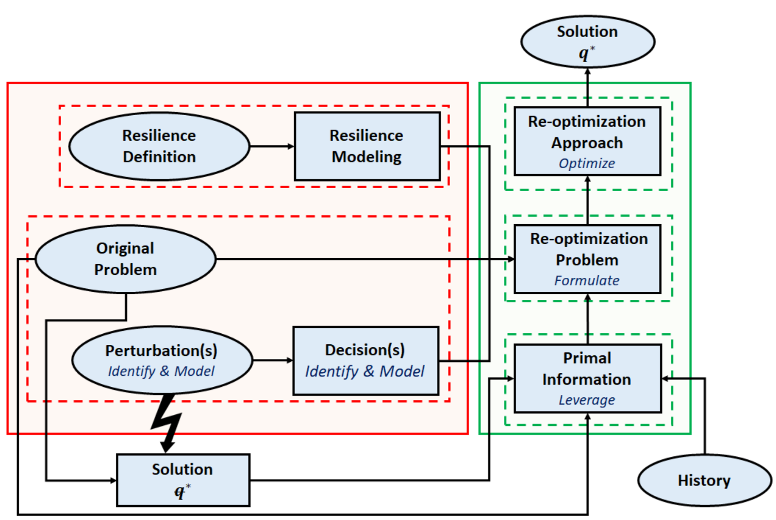


Figure 1: RRO framework

3.2 Managerial discussion

We discuss the RRO framework from a managerial perspective based on the SCOR model in Figure 2. A company having a set of suppliers and customers faces several perturbations. These perturbations can happen in any pillar of the SCOR model. Based on the point of emergence in the supply chain, these perturbations can classify into two types: internal and external. The internal ones are taking place inside the company. The external ones are either inbound (i.e., from the frontier with the supplier and above) or outbound (i.e., from the frontier with the customer and below). For instance, on the *make* pillar of the company, the company may face a machine breakdown curbing production. On the *deliver* pillar of the supplier, the company may receive raw materials later than planned. On the *source* pillar of the customer, the customers may cancel orders and or change their requirements.

Figure 2: SCOR model from AIMS

These perturbations also vary in terms of impact magnitude. Based on the impact, we distinguish minor and major perturbations. For instance, delayed delivery of raw materials is a minor perturbation when the company has enough safety stock. A machine breakdown curbing the production process is a major perturbation. It is worth mentioning that we focus on perturbations. The reason is that, after a perturbation, it is possible to recover quickly. This is not the case for disruptions. For instance, after a natural disaster (e.g., tsunami, or earthquake), a company may need several weeks to recover and resume operations.

To deal with these perturbations, the company has to take a set of actions to deal with them. For instance, on the company's *source* pillar, a company may decide to diversify suppliers or increase the safety stock levels. On the company's *make* pillar, the company may opt for a strict preventive maintenance strategy to diminish the machine breakdown occurrence. On the company's *deliver* pillar, the company may classify customer orders into confirmed and unconfirmed ones. Then, it can focus on producing and delivering the confirmed ones and postpone unconfirmed ones until confirmation.

The company must also seek resilience to adapt quickly to changes. To do so, it must establish a clear resilience definition. Then, it can develop an indicator that models resilience. Using these inputs, the company can design a re-optimization problem that can, when solved quickly based on the company's history, support decision-making when faced with perturbations. By bringing resilience and re-optimization together, the RRO framework ensures the following benefits. First, the re-optimization from a previous solution q^* (leveraging primal information) is quicker than re-optimizing from scratch. This ensures a quick solution, which implies quick adaptation, and thus operations continuity. Second, the consideration of resilience implies fewer changes. This makes the different involved stakeholders (e.g., operators, supervisors, and managers) from the company less worried about the changes to be made. A solution that requires fewer changes is always preferred by operators since they do not have to shift a lot from the way they used to work and plan.

In what follows, we illustrate the RRO framework considering a global SC. The red-framed part (*Problem Definition*) is Section 4 with the dashed red frames being subsections. The green-framed part (*Solution Methodology*) is Section 5 with the dashed green frames being subsections.

4 Problem definition

In this section, we present one implementation of the RRO framework for a real-world large-scale optimization problem. We first present the context (*Original Problem*, *Perturbations*, and *Decisions*). Then, we discuss resilience (*Resilience Definition* and *Resilience Modeling*).

4.1 Context

To present the context, we first describe briefly the original problem. Then, we highlight the perturbations and the decisions to tackle them. While the work presented for illustration is inspired by a large-scale mining company, its findings can be transferred, adapted, and applied to tackle similar difficult and large-scale problems in other industries.

4.1.1 Original problem

We consider the global SC of *OCP Group*, one of the largest phosphate companies worldwide, holding 70% of the world's phosphate rock reserves (Summaries, 2021). It has branches in Morocco, Brazil, India, and other countries, and specializes in phosphate mining, production, and exportation. Phosphate products include raw phosphate, phosphoric acid, and phosphate fertilizers.

The company promotes *precision farming*, i.e., utilizing a unique fertilizer for a specific type of soil (Auernhammer, 2001). As a result, its number of products has increased from 3 to more than 30 in recent years. Its global SC, highlighted in Figure 3, is made up of four main components, through which 45 raw, semi-finished, and finished products flow. The phosphate rocks are extracted from the mine; then, these rocks are transported using trucks to a physical treatment facility where they undergo the washing and floating processes. The washed rocks are transported by a 187 km slurry pipeline to the coastal processing plant for chemical treatment. Several derivative products are processed through 32 various chemical processes. The final products are then stored in 29 large tanks before being supplied through conveyors to quays, where vessels of clients worldwide are loaded. The coastal processing factory, as well as the loading port, spreads over an area of 5 km^2. On average, 37.6 million metric tons of phosphate rock are processed each year, accounting for 31% of the phosphate world market share. The supply chain is connected through 102 conveyors and pipelines (*OCP Group*, 2023).

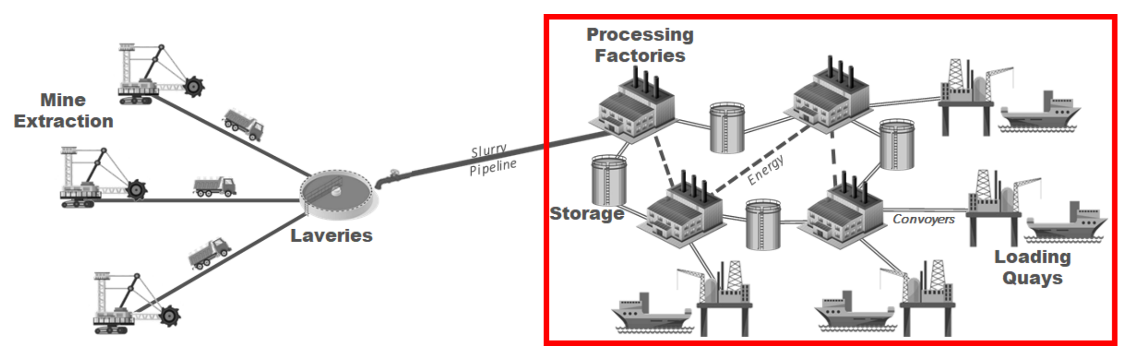


Figure 3: The phosphate supply chain

The considered large-scale optimization problem involves integrated production scheduling, inventory management, and vessel assignment (PSIMVA), grouping several components of the downstream supply chain (red-framed part in Figure 3), making it quite complex. More details about the PSIMVA can be found in Er Raqabi et al. (2023b). We are particularly dealing with perturbations happening on the port side, i.e., vessel assignment. In what follows, we assume that the company has already an optimal solution (schedule) $\mathfrak{q}^*$ obtained using, for example, solvers and OR techniques (Er Raqabi et al., 2023b; Himmich et al., 2023; Er Raqabi et al., 2023a). This problem is interesting because perturbations happen often, re-optimization from scratch is costly, and the company has enough historical data, which allows leveraging the primal information. It makes the problem suitable to illustrate the RRO framework presented in this paper.

4.1.2 Perturbations

OCP downstream operations are located in a coastal area. On the vessel assignment (VA) part of the PSIMVA, each customer vessel $v \in \mathcal{V} = \{1, ..., \overline{V}\}$ must be assigned to a quay $k \in \mathcal{K} = \{1, ..., \overline{K}\}$ for loading within a time interval (a few consecutive periods, with a period equivalent to one day). Let $\mathcal{T} = \{1, ..., \overline{T}\}$ be the set of $\overline{T}$ periods. The loading cannot be partial, i.e., either a vessel is fulfilled or not fulfilled. To accomplish this, a decision must be made on the assignment of each vessel $v \in \mathcal{V}$ to a quay based on a set of possible assignments $i \in \mathcal{I}_v = \{1, ..., \overline{I}_v\}$. We denote $\mathcal{I} = \bigcup_{v \in \mathcal{V}} \mathcal{I}_v$ and $\mathcal{I}_{vk}$ the set of possible assignment of vessel $v \in \mathcal{V}$ restricted to quay $k \in \mathcal{K}$. Each $i \in \mathcal{I}$ is a quadruplet $(\underline{t}, \overline{t}, k, v)$ where $i_1 = \underline{t}$ is the starting period, $i_2 = \overline{t}$ is the ending period, $i_3 = k$ is the quay, and $i_4 = v$ is the vessel. Each vessel has then a set of binary variables q_i with $i \in \mathcal{I}$, from which only one must be selected. OCP Group faces several perturbations related to the port, i.e., the VA part. We consider two perturbations defined below.

Definition 1. Weather Perturbation p_1 occurs when the weather in the port is bad enough to affect the normal operations.

Definition 2. Vessel Perturbation p_2 occurs when a vessel's arrival at the port is delayed.

If a perturbation p_1 occurs, all the vessels previously scheduled for loading during the perturbation's period(s) can no longer be loaded. Also, if a perturbation p_2 occurs for a given vessel, this vessel cannot be loaded as scheduled. Thus, these perturbations make $\mathfrak{q}^*$ no longer feasible. Furthermore, while being both minor, it is worth mentioning that the two perturbations considered are different in terms of time. In fact, assuming that the weather is accurately predicted a week before, perturbations p_1 can be tackled on a weekly basis, i.e., we eliminate the case where a vessel $v \in \mathcal{V}$ is in the loading process when a perturbation p_1 happens. Thus, after forecasting a weather perturbation period(s), vessels are rescheduled for loading with no intersection with the perturbation's period(s). On the other hand, perturbations p_2 happen in real-time.

From a modeling perspective, both perturbations p_1 and p_2 imply the removal of vessel assignment variables from the optimization model. For instance, if the weather is bad during a time interval $[t_1, t_2]$, all the assignment variables q_i, $i \in \mathcal{I}$ having a non-empty intersection with this time interval ($[i_1, i_2] \cap [t_1, t_2] \neq \emptyset$) are removed from the optimization model. Similarly, if a vessel $v \in \mathcal{V}$ is delayed, all its variables q_i, $i \in \mathcal{I}_v$ before its new arrival period are removed from the optimization model.

4.1.3 Decisions

When perturbations occur, decisions must be taken to recover quickly. Before introducing decisions, we present the following mild assumptions.

Assumption 1. The company has enough stocking space at the port.

Assumption 1 is realistic in the large-scale context because companies usually manage large product quantities. Thus, by design, they have quite large stocking spaces and entities.

Assumption 2. Customer vessels may arrive earlier than their schedule and wait close to the port.

Assumption 2 is realistic since in many ports worldwide (e.g., Singapore port in Figure 4), vessels wait close to the port until authorized to enter for loading.

Figure 4: Singapore port queue

Assumption 1 allows the company to produce and stock, and load after a delayed vessel arrives. This is very relevant and practical since it allows the company to maintain the production schedule as it is. Assumption 2 allows the possibility to advance some vessels (queuing close to the port) ahead of their schedules if others are delayed. We introduce now the two decisions taken to face perturbations.

Definition 3. A delay (del) decision d_1 is taken when the weather at the port is bad or when a vessel is delayed.

Definition 4. Let us consider two vessels v_1 and v_2 in $\mathcal{V}$ with similar products. If v_1 is delayed and v_2 is queuing close to the port, an advance (adv) decision d_2 occurs to allow loading vessel v_2 ahead of schedule.

It is worth mentioning that if a vessel $v \in \mathcal{V}$ is not expected to arrive within the planning horizon, it is simply delayed using decision d_1 beyond the planning horizon. Combining decisions d_1 and d_2, we make the following observation.

Observation 1. Swapping or permuting two vessels $v_1, v_2 \in \mathcal{V}$ with v_1 being ahead of v_2 is equivalent to applying decision d_1 to vessel v_1 and decision d_2 to vessel v_2.

Furthermore, for decisions d_1 and d_2, the whole space of feasible solutions is generated as shown in Proposition 1 below.

Proposition 1. Any feasible schedule $\bar{q}$ can be reached from the no longer feasible solution $\cancel{q}^*$ using decisions d_1 and d_2.

Proof. Consider a schedule $\bar{q}$. Delay beyond scheduling horizon using decision d_1 all vessels belonging to schedule $\cancel{q}^*$ and not to $\bar{q}$. If vessel $v \in \mathcal{V}$ is loaded earlier in schedule $\cancel{q}^*$ than $\bar{q}$, use decision d_1 to delay it. Similarly, if vessel $v \in \mathcal{V}$ is loaded later in schedule $\cancel{q}^*$ compared to schedule $\bar{q}$, use decision d_2 to advance it. $\square$

Proposition 1 is relevant because it ensures that the whole feasible space is explored. Thus, no feasible solution is discarded, including the new feasible schedule q^* we are looking for.

When a perturbation happens, we enumerate the set of delayed (resp. advanced) vessels $\mathcal{V}^{del}$ (resp. $\mathcal{V}^{adv}$). From a modeling perspective, decision d_1 corresponds to adding new variables q_i^{del}, $i \in \mathcal{I}_v$ to any delayed vessel $v \in \mathcal{V}^{del}$ with $i_1 \geq$ *loading*, *loading* being the potential (delayed) loading start period of vessel v. Decision d_2 corresponds to adding new variables q_i^{adv}, $i \in \mathcal{I}_v$ to any advanced vessel $v \in \mathcal{V}^{adv}$ with $i_1 \geq loading$, $loading$ being the potential (advanced) loading start period of vessel v. For advanced vessels, we keep also their initial assignment variables. To differentiate, we refer to these variables as q_i^{ini}, $i \in \mathcal{I}_v$ with $v \in \mathcal{V}^{adv}$.

4.2 Resilience

This section defines resilience and models it mathematically for our context.

4.2.1 Resilience definition

Among many definitions (Barroso et al., 2015), resilience can be defined as the ability of a system (e.g., company, organization, SC) to return to its original state or move to a new, more desirable state after being disturbed (Christopher and Peck, 2004).

In our context, SCR is the ability to recover or reach a better schedule while remaining as *close* as possible to the original schedule. By *close*, we imply fulfilling as many vessels as the previously optimal schedule while maintaining the least distance possible to the previously optimal schedule.

4.2.2 Resilience modeling

As per the definition above, we highlight two resilience indicators. The first is maximizing the total fulfillment (TF) and minimizing the distance between schedules (ΔD). The first indicator TF is the first KPI in Er Raqabi et al. (2023b). Denoting Q_v^p as the quantity of product $p \in \mathcal{P}$ ordered by vessel $v \in \mathcal{V}$, TF can be modeled as follows:

$$TF(q) = 100 \times \frac{\sum_{v \in \mathcal{V}, p \in \mathcal{P}_v} \sum_{i \in I_v} Q_v^p q_i}{\sum_{v \in \mathcal{V}, p \in \mathcal{P}_v} Q_v^p}$$

The second indicator ΔD can be modeled as follows:

$$\Delta D(q) = -100 \times \frac{\sum_{v \in \mathcal{V}^{del}, p \in \mathcal{P}_v} \sum_{i \in I_v} Q_v^p q_i^{del} + \sum_{v \in \mathcal{V}^{adv}, p \in \mathcal{P}_v} \sum_{i \in I_v} Q_v^p q_i^{adv}}{\sum_{v \in \mathcal{V}^{del} \bigcup \mathcal{V}^{adv}, p \in \mathcal{P}_v} Q_v^p}$$

For a given schedule q, $\Delta D(q)$ measures the percentage of vessels delayed and advanced. The negative sign is added since we want to maximize resilience and minimize the distance.

After modeling these two conflicting indicators, resilience can be modeled as a weighted objective:

$$R(q) = \alpha_1 \times TF(q) + \alpha_2 \times \Delta D(q),$$

where

$$(\alpha_1, \alpha_2) \in \mathbb{R}^2 \text{ such that } \alpha_1 + \alpha_2 = 1$$

In the next section, we develop the solution methodology. The objective is to find a feasible schedule that maximizes resilience under port constraints.

5 Solution methodology

In this section, we formulate the *Re-optimization Problem*, we highlight different ways to leverage *Primal Information*. Then, we provide the *Re-optimization Approach*.

5.1 Re-optimization problem

Following the design in Section 4, we have the following re-optimization problem constraints:

$$\sum_{i \in I_v} q_i^{del} \leq 1 \qquad \forall v \in \mathcal{V}^{del} \tag{1}$$

$$\sum_{i \in I_v} q_i^{ini} + q_i^{adv} \leq 1 \qquad \forall v \in \mathcal{V}^{adv} \tag{2}$$

$$\sum_{i \in I_v} q_i \leq 1 \qquad \forall v \in \mathcal{V} \setminus \mathcal{V}^{del} \cup \mathcal{V}^{adv} \tag{3}$$

$$\sum_{v \in \mathcal{V}^{del}} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t \leq i_2}} \mathbf{L}_v q_i^{del} + \sum_{v \in \mathcal{V}^{adv}} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t \leq i_2}} \mathbf{L}_v (q_i^{ini} + q_i^{adv}) + \sum_{v \in \mathcal{V} \setminus \mathcal{V}^{del} \bigcup \mathcal{V}^{adv}} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t \leq i_2}} \mathbf{L}_v q_i \leq \mathbf{L}_k \qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \tag{4}$$

$$\sum_{v \in \mathcal{V}, p \in \mathcal{P}_v} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t' \leq i_2}} \sum_{t'=t}^{t+6} \mathbf{Q}_v^p q_i \leq \mathbf{MAX} \qquad \forall k \in \mathcal{K}, t \in \mathcal{T}^- \tag{5}$$

$$q_i \in \mathbb{B} \qquad \forall i \in I_v, v \in V \tag{6}$$

Constraints (1) ensure that, at most, a single assignment is selected for each delayed vessel. Constraints (2) ensure that, at most, a single assignment is selected for each advanced vessel. Constraints (3) ensure that normal vessels must be fulfilled. Constraints (4) (to be discussed further below) control the respect to quay length restrictions. The length of a quay is noted $\mathbf{L}_k$ with $k \in \mathcal{K}$ while the length of a vessel is noted $\mathbf{L}_v$ with $v \in \mathcal{V}$. Constraints (5) are operational rules that control the maximum quantity ($\mathbf{MAX}$) that can be loaded on a given quay over a week. We note $\mathcal{T}^- = \mathcal{T} \setminus \{\overline{\mathcal{T}} - 5, ..., \overline{\mathcal{T}}\}$. Constraints (6) ensure the binary restrictions on the q variables.

By defining Ω as the set of constraints, the re-optimization problem is written as:

$$\begin{aligned} \max \quad & R(q) \\ \text{s.t.} \quad & q \in \Omega \end{aligned} \tag{Re-Opt}$$

5.2 Primal information

There are many ways to leverage primal information using the company's history, the previously optimal solution $\bar{q}^*$, and the original problem formulation.

5.2.1 Fixing

Fixing is the first option to leverage primal information. In fact, for unaffected (by perturbation(s)) vessels $v \in \mathcal{V} \setminus \mathcal{V}^{del} \cup \mathcal{V}^{adv}$, it is possible to fix them as per the previously optimal solution $\bar{q}^*$. In such a case, the corresponding term in the objective function becomes constant and we can remove constraints (3). Constraints (4) and (5) are written as follows:

$$\sum_{v \in \mathcal{V}^{del}} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t \leq i_2}} \mathbf{L}_v q_i^{del} + \sum_{v \in \mathcal{V}^{adv}} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t \leq i_2}} \mathbf{L}_v (q_i^{ini} + q_i^{adv}) \leq \mathbf{L}_k - \sum_{v \in \mathcal{V} \setminus \mathcal{V}^{del} \bigcup \mathcal{V}^{adv}} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t \leq i_2}} \mathbf{L}_v \bar{q}_i^* \qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \tag{7}$$

$$\sum_{v \in \mathcal{V}^{del} \bigcup \mathcal{V}^{adv}, p \in \mathcal{P}_v} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t' \leq i_2}} \sum_{t'=t}^{t+6} \mathbf{Q}_v^p q_i \leq \mathbf{MAX} - \sum_{v \in \mathcal{V} \setminus \mathcal{V}^{del} \bigcup \mathcal{V}^{adv}, p \in \mathcal{P}_v} \sum_{\substack{i \in \mathcal{I}_{vk} \\ i_1 \leq t' \leq i_2}} \sum_{t'=t}^{t+6} \mathbf{Q}_v^p \bar{q}_i^* \qquad \forall k \in \mathcal{K}, t \in \mathcal{T}^- \tag{8}$$

Fixing allows alleviating the model by eliminating a significant portion of binary variables q_i as well as $|\mathcal{V} \setminus \mathcal{V}^{del} \cup \mathcal{V}^{adv}|$ constraints.

5.2.2 Warm-start

Warm-starting is the second option to leverage primal information. For advanced vessels, they can be warm-started using their optimal assignment in $\mathfrak{q}^*$.

Proposition 2. Under the fixing option, $q_i^0 = \begin{cases} 0 & i \in \mathcal{I}_v, v \in \mathcal{V}^{del} \\ 0 \text{ or } \mathfrak{q}_i^* & i \in \mathcal{I}_v, v \in \mathcal{V}^{adv} \end{cases}$ is a feasible solution to Problem Re-Opt.

Proof. The delayed vessels cannot be warm-started because their previous assignments are removed. Their new assignment q_i^{del}, $i \in \mathcal{I}_v$ $v \in \mathcal{V}^{del}$ can be initiated with a 0. For the advanced vessels, the new assignment q_i^{adv}, $i \in \mathcal{I}_v$ $v \in \mathcal{V}^{adv}$ can be initiated with a 0. Still, the advanced vessels have their q_i^{ini}, $i \in \mathcal{I}_v$ $v \in \mathcal{V}^{adv}$ variables (from the previous schedule) in Re-Opt model. These variables can be warm-started using their values in $\mathfrak{q}^*$. □

If fixing is not applied, the unaffected vessels can also be warm-started.

Proposition 3. Without the fixing option, $q_i^0 = \begin{cases} 0 & i \in \mathcal{I}_v, v \in \mathcal{V}^{del} \\ 0 \text{ or } \mathfrak{q}_i^* & i \in \mathcal{I}_v, v \in \mathcal{V}^{adv} \\ \mathfrak{q}_i^* & i \in \mathcal{I}_v, v \in \mathcal{V} \setminus \mathcal{V}^{del} \cup \mathcal{V}^{adv} \end{cases}$ is a feasible solution to Problem Re-Opt.

Proof. Adding to Proposition 2, without fixing, q_i, $i \in \mathcal{I}_h$ $v \in \mathcal{V} \setminus \mathcal{V}^{del} \cup \mathcal{V}^{adv}$ can be warm-started using their assignment in $\mathfrak{q}^*$. □

Warm-starting allows for accelerating the solving process.

5.2.3 Valid inequalities

Valid inequalities are the third option to leverage primal information. They allow the strengthening of the model and obtaining tighter relaxations. For our Re-Opt model, we add the following valid inequalities:

$$\sum_{i \in I_{vk}} q_i^{del} \leq 1 \qquad \forall v \in \mathcal{V}^{del}, k \in \mathcal{K} \tag{9}$$

$$\sum_{i \in I_{vk}} q_i^{ini} + q_i^{adv} \leq 1 \qquad \forall v \in \mathcal{V}^{adv}, k \in \mathcal{K} \tag{10}$$

$$\sum_{i \in I_{vk}} q_i \leq 1 \qquad \forall v \in \mathcal{V} \setminus \mathcal{V}^{del} \cup \mathcal{V}^{adv}, k \in \mathcal{K} \tag{11}$$

Constraints (9) are a decomposition of Constraints (1) to quays. Constraints (10) are a decomposition of Constraints (2) to quays. Constraints (11)are a decomposition of Constraints (3) to quays.

5.2.4 Machine learning

ML is the fourth option to leverage primal information and has been used intensively recently in OR (Bengio et al., 2021). It consists of learning from the company's history (e.g., the pool of optimal schedules, weather history in the port, and perturbations history). It also supports the capturing of hidden trends that can help in making the solving approach *quicker*. In this section, we present the target, the data, the features, and the network structure qualitatively. All experiments are kept for Section 6.

Target. The goal of our ML model can be stated as follows: Given a vessel $v \in \mathcal{V}^{del} \cup \mathcal{V}^{adv}$, estimate the probability y_{vk} that vessel v is assigned to a quay $k \in \mathcal{K}$. Given such probabilities, we can select the top κ-quays for each vessel $v \in \mathcal{V}^{del} \cup \mathcal{V}^{adv}$, κ being an integer parameter. The role of the ML model is alleviating model Re-Opt by selecting only the *promising* quays for each $v \in \mathcal{V}^{del} \cup \mathcal{V}^{adv}$ and thus reducing the number of binary q variables significantly. For a given vessel $v \in \mathcal{V}^{del} \cup \mathcal{V}^{adv}$, the target of our ML model is constructing a vector $Y_v \in \mathbb{R}^{|\mathcal{K}|}$ such that $Y_{v_k} = y_{vk}$. Then, ranking the elements of vector Y_v in decreasing order will allow us to extract the top κ-quays for vessel v.

Data & features. Throughout the years, OCP Group has accumulated several schedules (solutions) where various vessels are assigned to various quays. For many instances, optimal schedules are available, and for some difficult instances, near-optimal schedules are available. These solutions will be used to train, validate, and test the ML model.

Each vessel $v \in \mathcal{V}^{del} \cup \mathcal{V}^{adv}$ and quay $k \in \mathcal{K}$ have a feature vector X_{vk}, which contains the following features:

1. For each product $p \in \mathcal{P}$:
   (a) A binary feature indicating whether vessel v contains product p, denoted $f_{vp}^{(a)}$.
   (b) A numerical feature indicating the quantity of product p required by vessel v, denoted $f_{vp}^{(b)}$.
2. A numerical feature corresponding to the earliest arrival period of vessel v, denoted $f^{(2)}$.
3. A numerical feature corresponding to the latest arrival period of vessel v, denoted $f^{(3)}$.
4. A numerical feature corresponding to the average loading time of vessel v, denoted $f^{(4)}$.
5. A numerical feature corresponding to the ratio of vessel v length to quay k length, denoted $f^{(5)}$.
6. A categorical feature corresponding to the destination of vessel v, denoted $f^{(6)}$.

These features contain all the information related to a given vessel $v \in \mathcal{V}$ and quay $k \in \mathcal{K}$.

Network structure. Each feature vector (entry) has a relatively large number of features. Furthermore, we did not find a strong correlation between any single feature and the target, suggesting that achieving a high prediction accuracy may require an ML model that can combine the features in a non-trivial way. Neural networks are known to perform well with entries containing many features. A neural network is composed of several neurons (also called perceptions) arranged in layers (Goodfellow et al., 2016). The first layer is called the input layer, and each neuron of this layer represents one feature. The last layer is called the output layer and holds the prediction $y_{vk}, k \in \mathcal{K}$ for $v \in \mathcal{V}$. A neural network may also contain one or several intermediate layers, called hidden layers, in which case it is called a deep neural network (DNN). The neurons of one layer are generally connected to neurons of the next layer. When predicting an entry, the neurons of the input layer are initialized to the value of the entry's features. Those values are then propagated throughout the network to the output layer. Each neuron computes a weighted sum of its inputs and applies to the result an activation function, which introduces non-linearity in the model. This value is then transmitted to the neurons of the next layer. The weights and the parameters of the activation functions are adjusted in a training phase to achieve the best accuracy. We, therefore, train a DNN on the task of predicting the target of a new entry.

Following the four options which allow leveraging the primal information, we present next the re-optimization approach.

5.3 Re-optimization approach

By leveraging the primal information to alleviate, strengthen, and warm-start model Re-Opt, the goal of this section is to find the new optimal schedule(s), i.e.:

$$q^*_{new} = \underset{q \in \Omega}{\arg\max} \quad R(q)$$

Before going to the solution, we introduce some relevant definitions. The first definition (Definition 5) highlights SCR mathematically. Since the resilience formula is a bi-objective function, the second definition (Definition 6) highlights the notion of a Pareto-optimal schedule.

Definition 5. Let $TF(\not{q}^*)$ be the updated total fulfillment after the removal of all vessels $v \in \mathcal{V}$ delayed beyond the scheduling horizon. A supply chain is said resilient if there exists a schedule $\bar{q}$ such that $TF(\bar{q}) = TF(\not{q}^*)$ and $|\Delta D|$ is minimal (i.e., if $|\Delta D'| < |\Delta D|$ then $TF'(\bar{q}) < TF(\not{q}^*)$). We refer to schedule $\bar{q}$ as a resilient schedule.

Definition 6. If for any positive weights α_1 and α_2 such that $\alpha_1 + \alpha_2 = 1$, there exists a schedule $\bar{q} \in \Omega$ with the property:

$$R(q) \leq R(\bar{q}) \qquad\qquad \forall q \in \Omega$$

Then schedule $\bar{q}$ is a Pareto-optimal solution for model Re-Opt.

Next, we show that any schedule $\bar{q}$ achieving SCR is Pareto-optimal.

Proposition 4. A resilient schedule $\bar{q}$ is a Pareto-optimal schedule.

Proof. As per Definition 5, schedule $\bar{q}$ ensures that $\forall q \in \Omega$:

$$TF(q) \leq TF(\not{q}^*) = TF(\bar{q}) \text{ and } \Delta D(q) \leq \Delta D(\bar{q})$$

Given two positive weights α_1 and α_2 such that $\alpha_1 + \alpha_2 = 1$, we have:

$$R(q) = \alpha_1 \times TF(q) + \alpha_2 \Delta D(q) \leq \alpha_1 \times TF(\bar{q}) + \alpha_2 \times \Delta D(\bar{q}) = R(\bar{q}) \qquad \forall q \in \Omega \qquad \square$$

We refer to this schedule as the resilient schedule. The inverse is not necessarily correct since the choice of weights may generate non-resilient schedules. For instance, when choosing $\alpha_1 = 0$ and $\alpha_2 = 1$, the Pareto-optimal schedule will minimize the distance without fulfilling delayed vessels. To find a resilient schedule $\bar{q}$, the bi-objective function weights α_1 and α_2 must be tuned. Since model Re-Opt is convex (TF, ΔD, and Ω are convex), there exist appropriate positive weights, as suggested by the following theorem (Wierzbicki, 1986):

Theorem 1. If $R(\Omega)$ is convex and $\bar{q}$ is Pareto-optimal for model Re-Opt, then there exist positive weights α_1 and α_2 with the property:

$$\alpha_1 \times TF(q) + \alpha_2 \times \Delta D(q) \leq \alpha_1 \times TF(\bar{q}) + \alpha_2 \times \Delta D(\bar{q}) \qquad \forall q \in \Omega$$

After finding appropriate weights and since we seek *quick* solving, we use the incremental large neighborhood search ($\mathcal{ILNS}$) metaheuristic of Er Raqabi et al. (2023b). Briefly, $\mathcal{ILNS}$ takes a Re-Opt instance and iterates over four steps: the *Vessel Assignment* step, the *Problem Reduction* step, the *Solving* step, and the *Wrap-up* step. In the *Vessel Assignment* step, after partitioning the re-optimization time horizon into smaller time intervals (e.g., weeks), we assign each vessel to a time interval. Using these assignments, we reduce further the pool of binary variables related to vessel assignment in the *Problem Reduction* step. Then, we solve the reduced problem in the *Solving* step. We keep iterating over the time horizon, using previous solutions as a warm-start until completion. Before returning a solution, in case there are still unfulfilled vessels, we try to fulfill them in the *Wrap-up* step. Compared to the standard LNS, which does not work efficiently for very large-scale optimization problems, $\mathcal{ILNS}$ destroys only the part of the solution that can be improved. This is because we do not have time for backtracking in such a huge MILP problem. Thus, we break the problem down, solve it, fix part of the solution, and move forward to improve that solution further. Further details are available in Er Raqabi et al. (2023b).

We illustrate the usage of $\mathcal{ILNS}$ in Figure 5. Starting from a monthly schedule $\not{q}^*$ in green, we call $\mathcal{ILNS}$ on a two-stage approach. The first stage is the re-optimization after a weather perturbation p_1

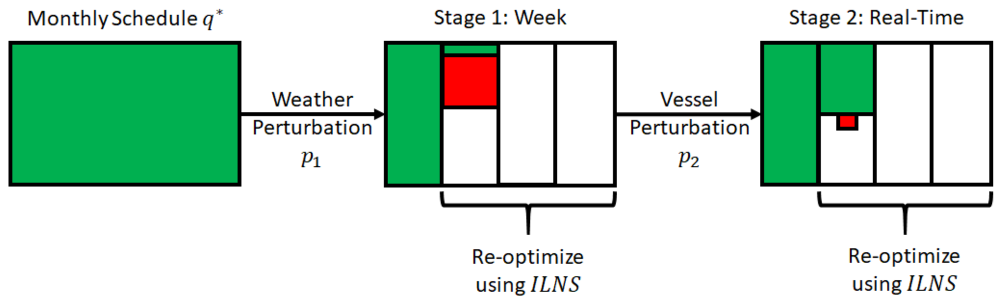


Figure 5: 2-stage $\mathcal{ILNS}$

(weekly basis). The second stage is the re-optimization after a vessel perturbation p_2 (real-time). For both stages, the schedule before the perturbation is maintained. The re-optimization using the two-stage $\mathcal{ILNS}$ is conducted on the periods that start from the perturbation period and continue until the end of the scheduling horizon. This is done by calling iteratively and when applicable, the $\mathcal{ILNS}$ just before the week begins for the weather perturbation p_1 and then whenever a vessel perturbation p_2 happens during the week. In what follows, we refer to the proposed approach as the 2-stage $\mathcal{ILNS}$.

6 Experimental design

To study whether the proposed approach is computationally efficient, we complement the analysis presented in previous sections with an extensive computational study. In this section, we describe the general characteristics of the test instances, the machine learning model (quantitatively), the computational setting, and the implementation details.

6.1 Instances

We pick six instances from Er Raqabi et al. (2023b). The features of these instances, including the time horizon (*Horizon*), the number of vessels (*Vessels*), the total demand (*Demand*) in tonne, and the total fulfillment ($\cancel{\mathrm{TF}}^*$) corresponding to solution $\cancel{\mathrm{q}}^*$ of each instance are presented in Table 3. For these instances, we know the optimal values as well as the optimal solutions.

Table 3: Instances

Name	Horizon	Vessels	Demand	$\cancel{\mathrm{TF}}^*$
I_1	32	54	806360	92.52%
I_2	32	58	826460	92.65%
I_3	32	39	1044550	93.34%
I_4	24	40	1043330	95.68%
I_5	32	61	1066290	93.10%
I_6	31	91	1304370	99.08%
Avg	31	57	1015227	94.40%

These instances will be perturbed according to three scenarios: perturbations p_1 alone, perturbations p_2 alone, and perturbations p_1 and p_2. From each instance above, we generate three perturbed instances. The first one has only p_1-type perturbations. The second one has only p_2-type perturbations. The third one has both p_1-type and p_2-type perturbations. The way we generate these instances is as follows. For p_1-type perturbations, we perturb a given % of periods with the scheduling horizon. For p_2-type perturbations, we perturb a given % of the considered vessels. For p_1-type and p_2-type perturbations, we perturb both periods and vessels. We refer to the given % as the perturbation percentage. An example is provided below.

Example 1. Let us consider an instance of a 2-week time horizon with ten vessels, two quays, and both p_1-type and p_2-type perturbations. With a perturbation percentage of 10% for both types, two periods out of 14 are randomly perturbed. Similarly, on the ten vessels, one vessel is randomly perturbed. This is visualized in red on the schedule in Figure 6. After perturbing periods 4 and 12, vessels with non-empty intersections with these two periods (i.e., v_2, v_4, v_6, v_8, and v_{10}) must be rescheduled. Similarly, for p_2-type perturbations, delayed vessel v_3 must also be rescheduled.

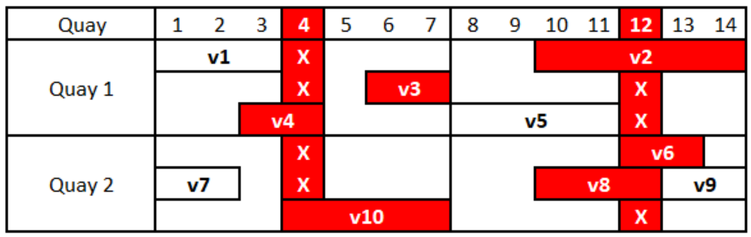


Figure 6: Example of a perturbed schedule

The re-optimizations are conducted in the order of appearance of perturbations following the 2-stage $\mathcal{ILNS}$ in Figure 5. Thus, starting with p_1-type perturbation in period 4, we reschedule vessels v_4 and v_{10}. This is the first re-optimization. Then, we re-optimize for a second time to reschedule v_3 (after p_2-type perturbation). Lastly, we re-optimize for a third time to reschedule vessels v_2, v_6, and v_8. The schedule obtained after completing all re-optimizations is referred to as the final schedule.

6.2 Machine learning model

As mentioned in the previous section, we use a ML model to construct Y_v for each vessel $v \in \mathcal{V}$. Our ML model is a DNN trained in a supervised training framework using data from several optimal solutions. To avoid confusion, we refer to these solutions as reference solutions. Each reference solution $\bar{q}$ from the history assigns several vessels to various quays. Thus, from a solution $\bar{q}$ with $|V_{\bar{q}}|$ vessels assigned, we can extract $|V_{\bar{q}}|$ pairs (X_{vk}, y_{vk}) where $v \in \mathcal{V}_{\bar{q}}$ and $k \in \mathcal{K}$. Furthermore, we can leverage the symmetry of the problem and collect all equivalent optimal reference solutions. Thus, we may have several possible assignments for each vessel $v \in V_{\bar{q}}$. Each X_{vk} is given a label y_{vk}. This label is computed as the frequency at which vessel $v \in V_{\bar{q}}$ is assigned to quay $k \in \mathcal{K}$ in all the considered reference solutions.

We train a DNN on the pool of reference solutions to predict the label of a new vector X_{vk}. We note y_{vk}^{prd} the predicted label of entry X_{vk}. The quay ranking is obtained by ordering the quays in decreasing order of y_{vk}^{prd}. Then, we can select the top κ-quays. The pairs (X_{vk}, y_{vk}) are split randomly into three disjoint subsets: a training set (60% of pairs), a validation set (20% of pairs), and a test set (20% of pairs).

The neural network is a feedforward fully connected DNN. Conforming with standard practices, the number of neurons in any hidden layer is less than or equal to that of the previous layer. All neurons except those of the input and output layers are rectilinear (ReLU) units. The output layer is composed of a single sigmoid unit so the output is in the interval $[0, 1]$. The hyperparameters and their possible values are given in Table 4. We determine the appropriate hyperparameters of the neural network and the training algorithm by performing a random grid search over the hyperparameter space. We select the model that achieves the best validation performance based on a performance indicator presented next.

A good performance indicator of the ML model can be obtained by taking the sum of the *true* labels in the top κ-quays in Y_v for all vessels. We use variable $j \in \{1, 2, ..., \kappa\}$ to refer to the ordered

Table 4: ML model hyperparameters

Name	Range	Type
Training algorithm	{SGD, Adam}	Categorical
Number of hidden layers	{2, 3, 4}	Integer
Neurons per layer	{10, 50, 150,..., 500}	Integer
Learning Rate	{0.0001, 0.001, 0.01, 0.01}	Float
Dropout	{0.1, 0.2, 0.3, 0.4}	Float

probabilities y_{vk_j} with y_{vk_1} being the top one and y_{vk_κ} the κ^{th} one. We measure the performance of our ML model by computing the label sum ratio in the top κ as follows:

$$TOP_\kappa = \frac{\sum_{v\in\mathcal{V}} \sum_{j=1}^{\kappa} y_{vk_j}}{\sum_{v\in\mathcal{V}} \sum_{k\in\mathcal{K}} y_{vk}}$$

Training is performed in a supervised fashion using either the stochastic gradient descent (SGD) (Bottou, 2010) or the Adam algorithm (Kingma and Ba, 2015). Several strategies are used to prevent overfitting. The neurons have a dropout probability between 0.1 and 0.4. Also, the validation performance (TOP_κ) is computed every 10 epochs and the training algorithm is stopped when it degrades twice in a row, or after a fixed number of epochs.

Training a neural network took less than 2 hours. Note that high training times are not a big concern for a real-world application because each ML model would be trained once and then used for several months or years. Finally, retraining a neural network is significantly faster than training one from scratch (minutes instead of hours) because the neural network is already in a near-optimal state and only needs to be slightly adjusted. The ML model performance is presented in Table 5. We note that, in our case, the loading port contains four quays. Thus, κ takes values from 1 to 4.

Table 5: Average TOP_κ for the test pairs

κ	1	2	3	4
Average TOP_κ	55.28%	95.73%	97.62%	100%

The ML model above will be used to alleviate the Re-Opt model. It is implemented in Python using the PyTorch library. All experiments were performed on a 40-core machine with 384 GB of memory.

6.3 Computational setting and implementation details

For the optimization part, the coding language is C++ and tests are conducted using version 12.10.0 of IBM ILOG CPLEX solver. All experiments were carried out on a 3.20GHz Intel(R) Core(TM) i7-8700 processor, with 64GiB System memory, running on Oracle Linux Server release 7.7. We use real-time to measure runtime. To make the implementations simple and easily replicable, we do not use any specialized codes or algorithms. Thus, we solve all MILPs using IBM ILOG CPLEX 12.10.0 run on a single thread.

We compare the following three methods:

- $\mathcal{ILNS}$: Solve the *Original Problem* in Er Raqabi et al. (2023b) from scratch using the $\mathcal{ILNS}$ metaheuristic as described in Er Raqabi et al. (2023b).
- MILP: Solve directly model Re-Opt with a MIP solver. In our case, we use the default CPLEX.
- 2-stage $\mathcal{ILNS}$: Solve using the proposed approach in Section 5.

The reason behind this choice of methods is the novelty of the problem, which was not tackled in the literature before, except in Er Raqabi et al. (2023b). Thus, to measure the solving performance, we compare our approach with the default CPLEX. Furthermore, given that we also want to compare the difference between optimizing from scratch (solving *Original Problem*) and re-optimizing from a given solution, we compare with the $\mathcal{ILNS}$ in Er Raqabi et al. (2023b).

7 computational results

In this section, we first find appropriate weights α_1 and α_2. Then, we compare the performance of the 2-stage $\mathcal{ILNS}$ against $\mathcal{ILNS}$ and MILP. After that, we conduct sensitivity analysis. We conclude with managerial insights.

7.1 Weights tuning

To find appropriate weights α_1 and α_2, we conduct a sensitivity analysis on instance I_1 with 10% p_1-type perturbations and 10% p_2-type perturbations. Since $\alpha_1 + \alpha_2 = 1$, we consider α_1 for the dichotomic search. Figure 7 shows the results for instance I_1, where $\alpha_1 = 0.73$ ensures the highest TF with the lowest $|\Delta D|$ possible.

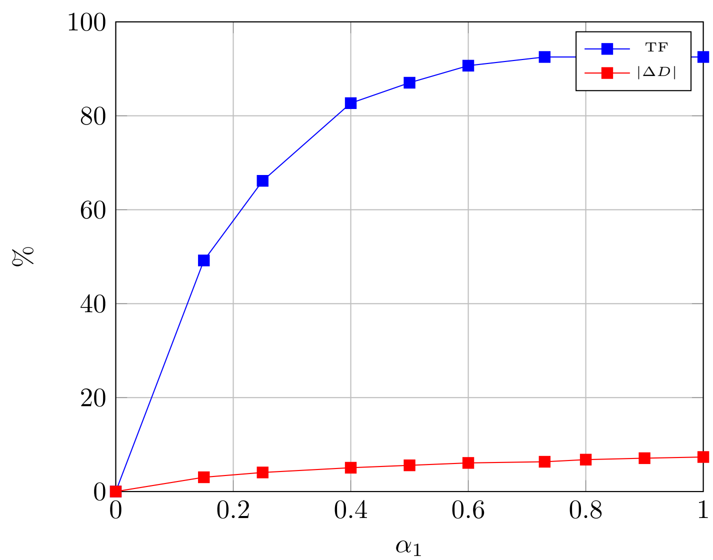


Figure 7: TF and $|\Delta D|$ evolution based on α_1 for instance I_1

Given that the instances are representative of the same problem, the suitable weights for I_1 are likely to be suitable for all instances. To confirm, we conducted a sensitivity analysis for other instances as well. Table 6 shows the results obtained.

As it can be observed, for all instances, the best α_1 value lies in the interval $[0.7, 0.8]$. Within this interval, all instances reach for the first time their optimal values. In what follows, we consider $\alpha_1 = 0.75$.

7.2 Performance comparison

We compare $\mathcal{ILNS}$, MILP, and 2-stage $\mathcal{ILNS}$. We report in Table 7 the perturbation percentage for both types (p_1 and p_2), the previously optimal TF ($\cancel{\text{TF}}^*$), the optimal TF obtained by each method (TF*), the optimal $|\Delta D|$ obtained by each method (ΔD^*), the optimal resilience (R^*), and the average

Table 6: Weight α_1 sensitivity analysis

Inst	α_1	0.00	0.15	0.25	0.40	0.50	0.60	0.73	0.80	0.90	1.00
I_1	TF	0.00%	49.20%	66.15%	82.69%	87.04%	90.67%	**92.52%**	92.52%	92.52%	92.52%
	$\|\Delta D\|$	0.00%	3.04%	4.06%	5.07%	5.58%	6.09%	6.34%	6.79%	7.10%	7.35%
I_2	TF	0.00%	49.69%	66.81%	83.52%	87.91%	91.31%	92.03%	**92.65%**	92.65%	92.65%
	$\|\Delta D\|$	0.00%	3.07%	4.10%	5.12%	5.63%	6.15%	6.44%	6.86%	7.17%	7.43%
I_3	TF	0.00%	50.18%	67.48%	84.34%	88.78%	92.48%	92.98%	**93.34%**	93.34%	93.34%
	$\|\Delta D\|$	0.00%	3.09%	4.11%	5.14%	5.66%	6.17%	6.51%	6.88%	7.20%	7.46%
I_4	TF	0.00%	51.17%	68.80%	86.00%	90.52%	94.30%	95.02%	**95.68%**	95.68%	95.68%
	$\|\Delta D\|$	0.00%	3.32%	4.42%	5.53%	6.08%	6.63%	6.91%	7.40%	7.74%	8.02%
I_5	TF	0.00%	50.09%	67.34%	84.18%	88.61%	92.30%	**93.10%**	93.10%	93.10%	93.10%
	$\|\Delta D\|$	0.00%	3.42%	4.56%	5.69%	6.26%	6.83%	6.98%	7.62%	7.97%	8.26%
I_6	TF	0.00%	61.50%	79.38%	89.31%	91.39%	92.48%	95.30%	**99.08%**	99.08%	99.08%
	$\|\Delta D\|$	0.00%	3.55%	4.74%	5.92%	6.51%	7.11%	7.15%	7.93%	8.29%	8.59%

time to re-optimize after each perturbation ($Time$). For perturbations, we consider three scenarios with 10% as a percentage as explained in Section 6.

As observed in Table 7, the three methods reach the optimal TF value, which is equal to the previously optimal value, except for instances I_2, I_3, I_4, and I_5 for which the optimal TF value decreases slightly under the scenario (10%,10%), i.e., 10% p_1-type perturbation and 10% p_2-type perturbation. Under this scenario, it becomes difficult to fulfill all vessels within the given time horizon and thus TF decreases. For the $Time$, the 2-stage $\mathcal{ILNS}$ outperforms both $\mathcal{ILNS}$ and MILP by factors of 44 and 15 on average, respectively. The $Time$ is affected more by p_1-type perturbations than p_2-type perturbations because the former involves more vessels for rescheduling at each re-optimization. On the resilience aspect, both MILP and 2-stage $\mathcal{ILNS}$ (which obtain the same results) outperform $\mathcal{ILNS}$. The latter re-optimizes the original problem in Er Raqabi et al. (2023b) from scratch without considering the resilience aspect. Thus, while the same optimal TF value is reached, the number of changes to the previously optimal schedule is larger. This is due to the changes in the production schedule, which is impacted by the perturbation. This change impacts significantly the vessels.

To analyze further the performance, we compare the schedules obtained by $\mathcal{ILNS}$ and 2-stage $\mathcal{ILNS}$ with the previously optimal schedule $\mathfrak{q}^*$. To do so, we consider instance I_1 under scenario (10%,10%) in Table 7. Figure 8 shows the previously optimal schedule with all perturbations in red.

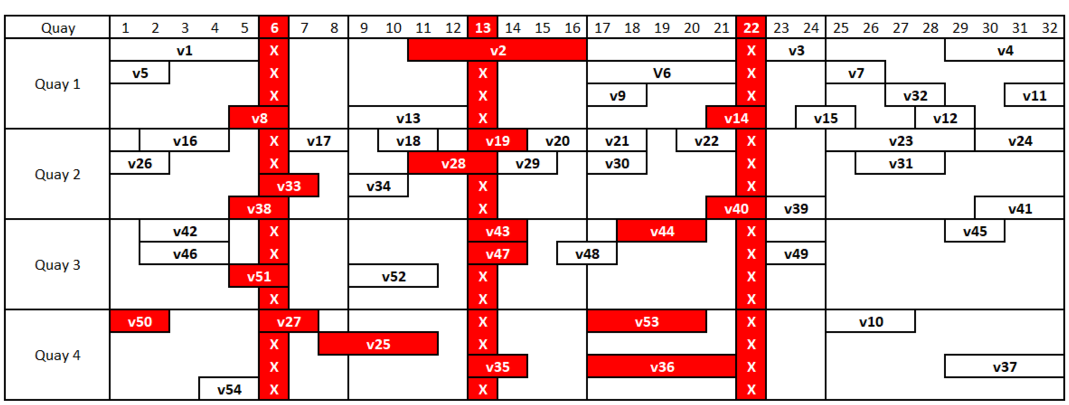


Figure 8: Perturbed instance I_1 under scenario (10%,10%)

Table 7: Performance comparison

p_1	p_2	$Inst$	~~TF~~*		$\mathcal{ILNS}$				MILP				2-stage $\mathcal{ILNS}$		
				TF^*	ΔD^*	R^*	$Time$	TF^*	ΔD^*	R^*	$Time$	TF^*	ΔD^*	R^*	$Time$
10%	-	I_1	92.52%	92.52%	53.60%	55.99%	518	92.52%	10.72%	66.71%	140	92.52%	10.72%	66.71%	10
		I_2	92.65%	92.65%	54.67%	55.82%	534	92.65%	10.93%	66.75%	168	92.65%	10.93%	66.75%	11
		I_3	93.34%	93.34%	55.77%	56.06%	550	93.34%	11.15%	67.22%	202	93.34%	11.15%	67.22%	13
		I_4	95.68%	95.68%	56.88%	57.54%	605	95.68%	11.38%	68.92%	242	95.68%	11.38%	68.92%	15
		I_5	93.10%	93.10%	62.28%	54.25%	635	93.10%	12.46%	66.71%	278	93.10%	12.46%	66.71%	16
		I_6	99.08%	99.08%	59.45%	59.45%	673	99.08%	11.89%	71.34%	305	99.08%	11.89%	71.34%	19
		Avg	**94.40%**	**94.40%**	**57.11%**	**59.45%**	**586**	**94.40%**	**11.42%**	**67.94%**	**222**	**94.40%**	**11.42%**	**67.94%**	**14**
-	10%	I_1	92.52%	92.52%	31.70%	61.47%	432	92.52%	6.34%	67.81%	98	92.52%	6.34%	67.81%	7
		I_2	92.65%	92.65%	32.20%	61.44%	454	92.65%	6.44%	67.88%	108	92.65%	6.44%	67.88%	8
		I_3	93.34%	93.34%	32.55%	61.87%	476	93.34%	6.51%	68.38%	119	93.34%	6.51%	68.38%	10
		I_4	95.68%	95.68%	33.85%	63.30%	500	95.68%	6.77%	70.07%	142	95.68%	6.77%	70.07%	11
		I_5	93.10%	93.10%	34.90%	61.10%	525	93.10%	6.98%	68.08%	185	93.10%	6.98%	68.08%	13
		I_6	99.08%	99.08%	35.75%	65.37%	561	99.08%	7.15%	72.52%	262	99.08%	7.15%	72.52%	16
		Avg	**94.40%**	**94.40%**	**33.49%**	**62.42%**	**491**	**94.40%**	**6.70%**	**69.12%**	**152**	**94.40%**	**6.70%**	**69.12%**	**11**
10%	10%	I_1	92.52%	92.52%	52.54%	56.26%	475	92.52%	15.01%	65.64%	119	92.52%	15.01%	65.64%	8
		I_2	92.65%	92.65%	54.64%	55.83%	494	92.65%	15.61%	65.58%	138	92.65%	15.61%	65.58%	9
		I_3	93.34%	90.54%	57.37%	53.56%	513	90.54%	16.39%	63.81%	160	90.54%	16.39%	63.81%	11
		I_4	95.68%	93.77%	59.63%	55.42%	552	93.77%	17.54%	65.94%	192	93.77%	17.54%	65.94%	13
		I_5	93.10%	90.31%	61.35%	52.39%	580	90.31%	18.59%	63.08%	232	90.31%	18.59%	63.08%	15
		I_6	99.08%	97.10%	64.14%	56.79%	617	97.10%	21.38%	67.48%	284	97.10%	21.38%	67.48%	17
		Avg	**94.40%**	**92.81%**	**58.28%**	**55.04%**	**538**	**92.81%**	**17.42%**	**65.26%**	**187**	**92.81%**	**17.42%**	**65.26%**	**12**

Figure 9 shows the final schedule obtained using the $\mathcal{ILNS}$ method. All the vessels in green were rescheduled to take into consideration all perturbations. As can be seen, when re-optimizing from scratch, several unperturbed vessels are re-scheduled as well. Of all 54 vessels, 22 vessels only remained as previously planned.

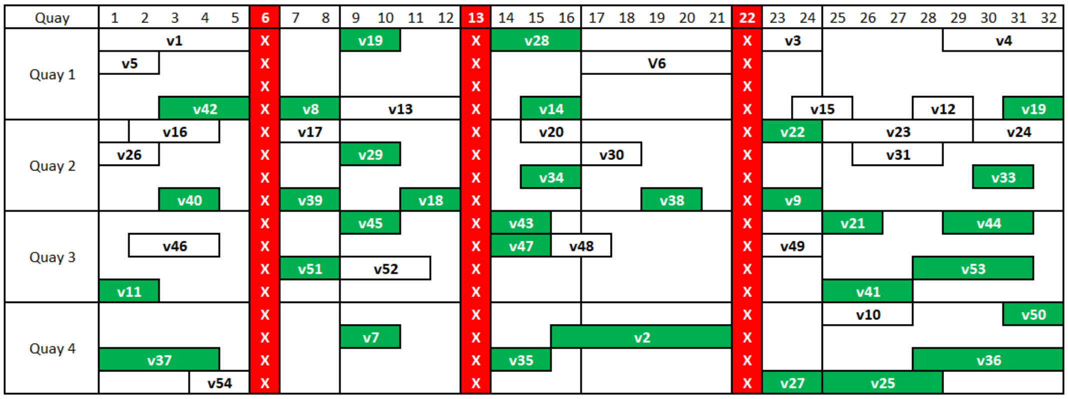


Figure 9: Final schedule obtained by $\mathcal{ILNS}$ for perturbed instance I_1 under scenario (10%,10%)

Figure 10 shows the final schedule obtained when re-optimizing while considering the resilience aspect. We observe that the 2-stage $\mathcal{ILNS}$ re-schedules only the perturbed vessels. Furthermore, it keeps several vessels close to their original schedule. For instance, vessel v_{27} in Quay 4 was moved to periods 7 and 8 from periods 6 and 7 on the same quay, i.e., the period following the bad weather period. Out of all 54 vessels, only the 18 vessels perturbed were rescheduled.

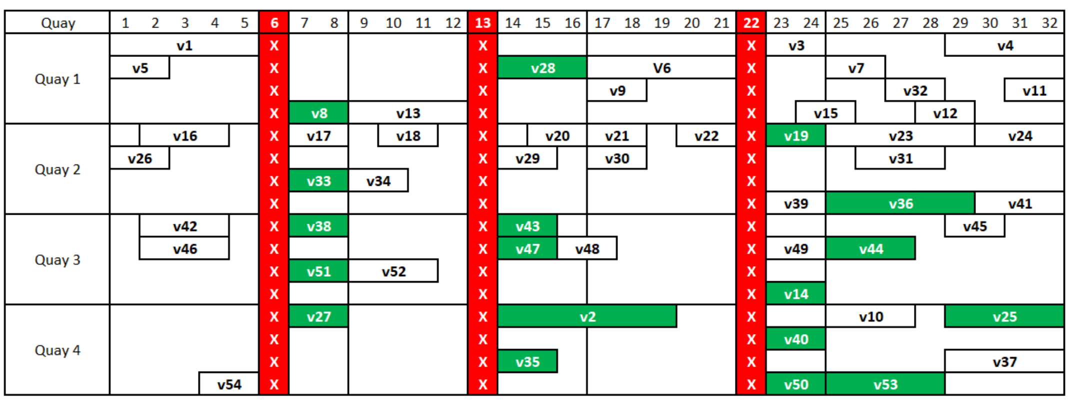


Figure 10: Final schedule obtained by 2-stage $\mathcal{ILNS}$ for perturbed instance I_1 under scenario (10%,10%)

When observing the schedules in Figures 8, 9, and 10, one can infer that, when dealing with perturbations, optimizing locally can be more relevant than optimizing globally to achieve resilience, especially when the optimal value can be reached and the global problem is a large-scale optimization problem. Local optimization explores just the affected part of the mathematical model and tries to correct it while global optimization might correct the affected part of the mathematical model while making changes to unaffected parts. In our case, when re-optimizing from scratch, $\mathcal{ILNS}$ changes the production schedule, which in turn affects more vessels beyond the perturbed ones.

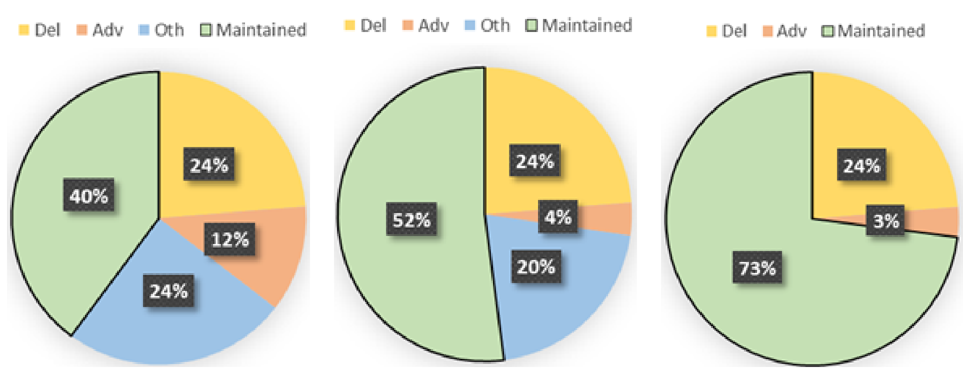
Table 8 shows the number of delayed (Del), advanced (Adv), and other (Oth) vessels rescheduled in the final schedules compared to the initial ones. As it can be observed, the 2-stage $\mathcal{ILNS}$ reschedules

Table 8: Number of delayed, advanced, and other vessels rescheduled in the final schedule

p_1	p_2	Inst	Horizon	Vessels	$\mathcal{ILNS}$			MILP			2-stage $\mathcal{ILNS}$		
					Del	Adv	Oth	Del	Adv	Oth	Del	Adv	Oth
10%	-	I_1	32	54	13	5	9	13	1	11	13	0	0
		I_2	32	58	14	4	9	14	2	12	14	0	0
		I_3	32	39	11	3	11	11	0	8	11	0	0
		I_4	24	40	12	6	12	12	3	8	12	0	0
		I_5	32	61	15	11	18	15	4	12	15	0	0
		I_6	31	91	18	15	27	18	5	18	18	0	0
		Avg	**31**	**57**	**14**	**7**	**14**	**14**	**1**	**11**	**14**	**0**	**0**
-	10%	I_1	32	54	5	4	8	5	0	9	5	0	0
		I_2	32	58	6	3	8	6	0	9	6	0	0
		I_3	32	39	4	2	10	4	0	6	4	0	0
		I_4	24	40	4	5	10	4	1	6	4	0	0
		I_5	32	61	6	9	15	6	2	10	6	0	0
		I_6	31	91	9	12	22	9	4	15	9	0	0
		Avg	**31**	**57**	**6**	**6**	**12**	**6**	**1**	**9**	**6**	**0**	**0**
10%	10%	I_1	32	54	18	4	10	18	0	14	18	0	0
		I_2	32	58	22	8	10	22	5	15	22	7	0
		I_3	32	39	17	0	13	17	3	10	17	0	0
		I_4	24	40	18	4	13	18	3	10	18	4	0
		I_5	32	61	23	11	19	23	5	16	23	9	0
		I_6	31	91	30	15	28	30	7	24	30	14	0
		Avg	**31**	**57**	**21**	**7**	**16**	**21**	**4**	**15**	**21**	**6**	**0**

mainly the affected vessels. In tight scenarios, it reschedules advanced vessels to free room for delayed ones. The other vessels are kept as initially scheduled because of the fixing strategy. On the other side, $\mathcal{ILNS}$ reschedules significantly more vessels (Oth) unaffected by perturbations and advanced vessels. The same observation holds for MILP because there is no fixing.

We show in Figure 11 the average percentage of vessels rescheduled by each of the methods in the final schedule(s). It includes delayed, advanced, and other vessels. The vessels *Maintained* are the ones that were maintained as per the initial schedule(s).



(a) $\mathcal{ILNS}$ **(b) MILP** **(c) 2-stage** $\mathcal{ILNS}$

Figure 11: Average percentage of vessels delayed, advanced, and others rescheduled in the final schedule(s). Maintained vessels are vessels kept as per the previously optimal solution(s)

Figure 11 highlights that the percentage of maintained vessels varies significantly among the three methods. When optimizing globally and considering the original problem, production is rescheduled leading to the rescheduling of around 60% of the vessels. The MILP reschedules the delayed vessels and some vessels that can be advanced. Still, it reschedules other vessels, which are not affected by perturbations. Compared to $\mathcal{ILNS}$, the MILP keeps 12% more vessels as previously planned. For the 2-stage $\mathcal{ILNS}$, only the vessels delayed are rescheduled with some advanced vessels (to free space). This shows, that when optimizing locally with resilience taken into consideration and primal information leveraged, the obtained schedules are closer to the initially planned ones.

7.3 Sensitivity analysis

We conduct a sensitivity analysis on the perturbation percentage. We consider instance I_1 and vary the perturbation percentage from 5% to 25%. Table 9 shows the obtained results. We observe that the higher the perturbation percentage, the lower the resilience value. Indeed, with more perturbations, the TF value is more likely to decrease because there is less room to fulfill all vessels, and the ΔD^* value increases because there are more movements. In Table 9, the TF^* value decreases starting from a perturbation percentage of 20%.

Table 9: Perturbation percentage impact on instance I_1

p_1	p_2	$\cancel{\mathrm{TF}}^*$	$\mathcal{ILNS}$			MILP			2-stage $\mathcal{ILNS}$		
			TF^*	ΔD^*	R^*	TF^*	ΔD^*	R^*	TF^*	ΔD^*	R^*
5%	-	92.52%	92.52%	42.88%	58.67%	92.52%	8.58%	67.25%	92.52%	8.58%	67.25%
-	5%	92.52%	92.52%	25.36%	63.05%	92.52%	5.07%	68.12%	92.52%	5.07%	68.12%
5%	5%	92.52%	92.52%	42.03%	58.88%	92.52%	12.01%	66.39%	92.52%	12.01%	66.39%
10%	-	92.52%	92.52%	53.60%	55.99%	92.52%	10.72%	66.71%	92.52%	10.72%	66.71%
-	10%	92.52%	92.52%	31.70%	61.47%	92.52%	6.34%	67.81%	92.52%	6.34%	67.81%
10%	10%	92.52%	92.52%	52.54%	56.26%	92.52%	15.01%	65.64%	92.52%	15.01%	65.64%
15%	-	92.52%	92.52%	64.32%	53.31%	92.52%	12.86%	66.17%	92.52%	12.86%	66.17%
-	15%	92.52%	92.52%	38.04%	59.88%	92.52%	7.61%	67.49%	92.52%	7.61%	67.49%
15%	15%	92.52%	92.52%	63.04%	53.63%	92.52%	18.01%	64.89%	92.52%	18.01%	64.89%
20%	-	92.52%	89.76%	75.04%	48.56%	89.76%	15.01%	63.57%	89.76%	15.01%	63.57%
-	20%	92.52%	90.67%	44.38%	56.91%	90.67%	8.88%	65.78%	90.67%	8.88%	65.78%
20%	20%	92.52%	88.87%	73.55%	48.26%	88.87%	21.01%	61.40%	88.87%	21.01%	61.40%
25%	-	92.52%	87.07%	82.54%	44.67%	87.07%	16.51%	61.18%	87.07%	16.51%	61.18%
-	25%	92.52%	87.95%	48.82%	53.76%	87.95%	9.76%	63.52%	87.95%	9.76%	63.52%
25%	25%	92.52%	86.20%	80.90%	44.42%	86.20%	23.12%	58.87%	86.20%	23.12%	58.87%

We also check the impact of the primal information. To do so, we run additional tests considering no warm start ($NoWS$), no fixing ($NoFix$), no valid inequalities ($NoVI$), and no machine learning ($NoML$). We report the results with all options (All), which are the same as instance I_1 results on Table 7, for comparison purposes.

While warm starting and valid inequalities do not impact the $Time$ a lot, the fixing and machine learning strategies impact significantly the $Time$. On average, $Time$ is reduced by a factor of 1.8 and 1.7 per perturbation when using fixing and machine learning options, respectively.

The ML model plays two roles. First, it computes assignment probabilities for each pair vessel and quay. Second, it identifies less promising quays for each vessel. Then, the corresponding q variables are removed from the mathematical model. It is worth mentioning that we could reach TF^* by selecting only the top 2-quays for each vessel. This is equivalent to reaching the optimal value when considering just 50% of the binary assignment variables for each vessel.

7.4 Managerial insights

As shown above, the 2-stage $\mathcal{ILNS}$ reinforced with primal information ensures both quick re-optimization and resilient schedules. The quick re-optimization provides a decision support system to decision makers (e.g., planners, managers, operators) and allows them to consider various what-if

Table 10: Primal information impact on $Time$ for instance I_1

p_1	p_2	$Inst$	~~TF~~*	$NoWS$	$NoFix$	$NoVI$	$NoML$	All
10%	-	I_1	92.52%	11	17	11	16	10
		I_2	92.65%	12	19	12	18	11
		I_3	93.34%	14	23	14	21	13
		I_4	95.68%	17	27	17	25	15
		I_5	93.10%	18	29	18	27	16
		I_6	99.08%	20	33	20	31	19
		Avg	**94.40%**	**15**	**25**	**15**	**23**	**14**
-	10%	I_1	92.52%	7	12	7	11	7
		I_2	92.65%	9	14	9	13	8
		I_3	93.34%	11	18	11	17	10
		I_4	95.68%	13	21	13	19	11
		I_5	93.10%	15	24	15	22	13
		I_6	99.08%	18	29	18	27	16
		Avg	**94.40%**	**12**	**20**	**12**	**18**	**11**
10%	10%	I_1	92.52%	9	15	9	14	8
		I_2	92.65%	10	17	10	16	9
		I_3	93.34%	13	21	13	19	11
		I_4	95.68%	15	24	15	22	13
		I_5	93.10%	16	26	16	25	15
		I_6	99.08%	19	31	19	29	17
		Avg	**94.40%**	**14**	**22**	**14**	**21**	**12**

scenarios and cases. The resilient schedules permit to stay as close as possible to the previously planned schedule, thus involving fewer changes and sustaining operators' satisfaction.

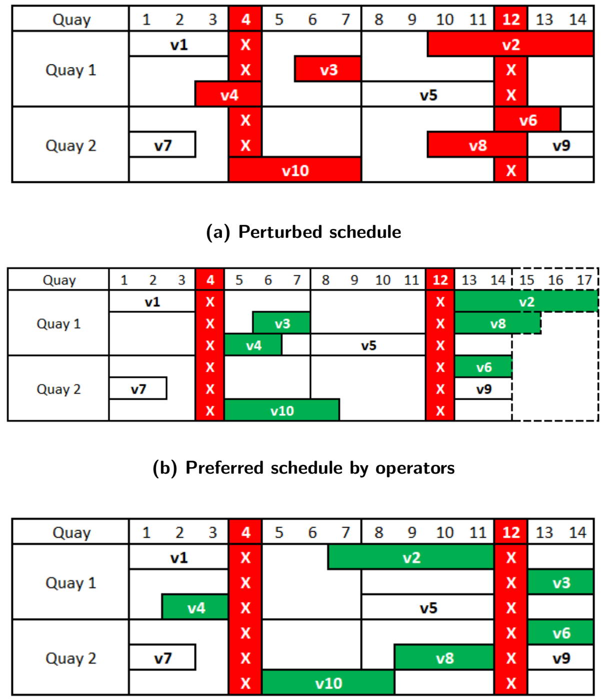
Another interesting aspect is the advantage of local optimization. Indeed, when the perturbation is local, optimizing locally is better than optimizing globally, especially when the former reaches the optimal solution. First, local optimization involves a smaller problem. Second, it changes only the part of the problem affected by the perturbation. Third, it leverages the available primal information.

We distinguish two types of schedules from which decision-makers can select. The first schedule delays all the perturbed vessels, as shown in the example in Figure 12b. Operators prefer this schedule because it involves changes that require less effort. Still, it may imply going beyond the planning time horizon. In Figure 12b, three periods are added to fulfill vessels v_2 and v_8. This may decrease customer satisfaction. The second schedule modifies the perturbed vessels and allows both delay and advance. This schedule is less preferred by operators compared to the first schedule because it involves working ahead of schedule as shown in Figure 12c. Still, it maintains customer satisfaction and may increase it if customers' vessels are fulfilled earlier than expected (e.g., vessel v_2). It also ensures that the planned vessels are fulfilled within the initial time horizon.

The ML helped us identify some hidden trends. For instance, it shows that some products are loaded on specific quays while others are loaded on other quays. An operational explanation is that liquid-type products are loaded on specific quays while solid-type products are loaded on different quays. The problem can then be decomposed by product type.

8 Conclusion

This research presents a generic and scalable resilience re-optimization framework. We highlight various ways of leveraging the primal information, including fixing, warm-start, valid inequalities, and machine learning. Using a real-world large-scale problem for illustration, we discuss uncertainties, confirm their impact on the model, and model recovery decisions, highlighting the need for resilience. Then, we model the re-optimization problem to maximize resilience. Finally, we conduct extensive computational



(a) Perturbed schedule

(b) Preferred schedule by operators

(c) Preferred schedule by customers

Figure 12: Two types of schedules

experiments to demonstrate the power of our proposed framework and solution methodology. The proposed framework is scalable and generic to any large-scale company facing several perturbations and seeking quick re-optimization and resilient solutions. Future work includes extending the framework to tackle disruptions such as earthquakes, tsunamis, and wars, and considering cases where local optimization is no longer enough (suboptimal solutions, infeasible solutions, etc.). In such cases, global optimization becomes a must.

References

Akkermans, H. and Van Wassenhove, L. N. (2018). Supply chain tsunamis: research on low-probability, high-impact disruptions. Journal of Supply Chain Management, 54(1):64–76.

An, S., Cui, N., Bai, Y., Xie, W., Chen, M., and Ouyang, Y. (2015). Reliable emergency service facility location under facility disruption, en-route congestion and in-facility queuing. Transportation Research Part E: Logistics and Transportation Review, 82:199–216.

Archetti, C., Guastaroba, G., and Speranza, M. G. (2013). Reoptimizing the rural postman problem. Computers & Operations Research, 40(5):1306–1313.

Auernhammer, H. (2001). Precision farming: The environmental challenge. Computers and Electronics in Agriculture, 30(1–3):31–43.

Barroso, A., Machado, V., Carvalho, H., and Machado, V. C. (2015). Quantifying the supply chain resilience. Applications of Contemporary Management Approaches in Supply Chains, 13:38.

Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d'horizon. European Journal of Operational Research, 290(2):405–421.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In 19th International Conference on Computational Statistics, COMPSTAT 2010, Paris, France, August 22-27, 2010 - Keynote, Invited and Contributed Papers, pages 177–186. Physica-Verlag.

Bruno, G., Cavola, M., Diglio, A., Laporte, G., and Piccolo, C. (2021). Reorganizing postal collection operations in urban areas as a result of declining mail volumes–a case study in Bologna. Journal of the Operational Research Society, 72(7):1591–1606.

Chakraborty, I. and Maity, P. (2020). COVID-19 outbreak: Migration, effects on society, global environment and prevention. Science of the Total Environment, 728:138882.

Chen, L. and Miller-Hooks, E. (2012). Resilience: an indicator of recovery capability in intermodal freight transport. Transportation Science, 46(1):109–123.

Chopra, S., Sodhi, M., and Lücker, F. (2021). Achieving supply chain efficiency and resilience by using multi-level commons. Decision Sciences, 52(4):817–832.

Christopher, M. and Peck, H. (2004). Building the resilient supply chain. International Journal of Logistics Management, 15(2):1–13.

Doerr, B., Doerr, C., and Neumann, F. (2019). Fast re-optimization via structural diversity. In Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '19, pages 233–241.

Dong, Y., Maravelias, C. T., and Jerome, N. F. (2018). Reoptimization framework and policy analysis for maritime inventory routing under uncertainty. Optimization and Engineering, 19:937–976.

D'Ariano, A., Albrecht, T., Allan, J., Brebbia, C., and Rumsey, A. (2010). Running time re-optimization during real-time timetable perturbations. Timetable Planning and Information Quality, 1:147–156.

Elluru, S., Gupta, H., Kaur, H., and Singh, S. P. (2019). Proactive and reactive models for disaster resilient supply chain. Annals of Operations Research, 283:199–224.

OCP Group (2023). OCP Group Morocco. https://www.ocpgroup.ma/. Accessed on 05.04.2023.

Er Raqabi, E. M., El Hallaoui, I., and Soumis, F. (2023a). The Primal Benders Decomposition. Les Cahiers du GERAD, G–2023–27.

Er Raqabi, E. M., Himmich, I., El Hachemi, N., El Hallaoui, I., and Soumis, F. (2023b). Incremental LNS framework for integrated production, inventory, and vessel scheduling: Application to a global supply chain. Omega, 116:102821.

Gliessman, S. (2022). Why is there a food crisis? Agroecology and Sustainable Food Systems, 46(9):1301–1303.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. MIT press.

Hasani, A., Mokhtari, H., and Fattahi, M. (2021). A multi-objective optimization approach for green and resilient supply chain network design: A real-life case study. Journal of Cleaner Production, 278:123199.

Hassani, R., Desaulniers, G., and Elhallaoui, I. (2020). Real-time personnel re-scheduling after a minor disruption in the retail industry. Computers & Operations Research, 120:104952.

Hassani, R., Desaulniers, G., and Elhallaoui, I. (2021). Real-time bi-objective personnel re-scheduling in the retail industry. European Journal of Operational Research, 293(1):93–108.

Himmich, I., Er Raqabi, E. M., El Hachemi, N., El Hallaoui, I., Metrane, A., and Soumis, F. (2023). MPILS: An Automatic Tuner for MILP Solvers. Computers & Operations Research, 159:106344.

Hosseini, S., Ivanov, D., and Dolgui, A. (2019a). Review of quantitative methods for supply chain resilience analysis. Transportation Research Part E: Logistics and Transportation Review, 125:285–307.

Hosseini, S., Morshedlou, N., Ivanov, D., Sarder, M., Barker, K., and Al Khaled, A. (2019b). Resilient supplier selection and optimal order allocation under disruption risks. International Journal of Production Economics, 213:124–137.

Kamalahmadi, M. and Parast, M. M. (2016). A review of the literature on the principles of enterprise and supply chain resilience: Major findings and directions for future research. International Journal of Production Economics, 171:116–133.

Khaled, A. A., Jin, M., Clarke, D. B., and Hoque, M. A. (2015). Train design and routing optimization for evaluating criticality of freight railroad infrastructures. Transportation Research Part B: Methodological, 71:71–84.

Khalili, S. M., Jolai, F., and Torabi, S. A. (2017). Integrated production–distribution planning in two-echelon systems: a resilience view. International Journal of Production Research, 55(4):1040–1064.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings.

Lee, J. M. Y. and Wong, E. Y. C. (2021). Suez Canal blockage: an analysis of legal impact, risks and liabilities to the global supply chain. In MATEC web of conferences, volume 339, page 01019. EDP Sciences.

Mbah, R. E. and Wasum, D. F. (2022). Russian-Ukraine 2022 War: A review of the economic impact of Russian-Ukraine crisis on the USA, UK, Canada, and Europe. Advances in Social Sciences Research Journal, 9(3):144–153.

Sahebjamnia, N., Torabi, S. A., and Mansouri, S. A. (2018). Building organizational resilience in the face of multiple disruptions. International Journal of Production Economics, 197:63–83.

Sawik, T. (2019). Two-period vs. multi-period model for supply chain disruption management. International Journal of Production Research, 57(14):4502–4518.

Schieber, B., Shachnai, H., Tamir, G., and Tamir, T. (2018). A theory and algorithms for combinatorial reoptimization. Algorithmica, 80:576–607.

Shen, H., Fu, M., Pan, H., Yu, Z., and Chen, Y. (2020). The impact of the COVID-19 pandemic on firm performance. Emerging Markets Finance and Trade, 56(10):2213–2230.

Summaries, M. C. (2021). Mineral commodity summaries. US Geological Survey: Reston, VA, USA, 200.

Wierzbicki, A. P. (1986). On the completeness and constructiveness of parametric characterizations to vector optimization problems. Operations-Research-Spektrum, 8(2):73–87.

Winn, M., Kirchgeorg, M., Griffiths, A., Linnenluecke, M. K., and Günther, E. (2011). Impacts from climate change on organizations: a conceptual foundation. Business Strategy and the Environment, 20(3):157–173.

Yang, T. and Fan, W. (2016). Information management strategies and supply chain performance under demand disruptions. International Journal of Production Research, 54(1):8–27.

Yang, Z., Aydın, G., Babich, V., and Beil, D. R. (2009). Supply disruptions, asymmetric information, and a backup production option. Management Science, 55(2):192–209.