Received 23 July 2023, accepted 12 August 2023, date of publication 15 August 2023, date of current version 22 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3305572

RESEARCH ARTICLE

Modeling the Bulk Port Belt-Conveyor Routing Problem Considering Interactions With Storage Spaces and Loading Operations

SARA MALLAH[1,2], AFAF ALOULLAL[3], OULAID KAMACH[1,2], MALEK MASMOUDI[4], KHALID KOUISS[2,5], AND AHMED CHEBAK[2]

[1]Laboratoire des Technologies Innovantes (LTI), National School of Applied Sciences, University Abdelmalek Essaadi, Tangier 90040, Morocco
[2]Green Tech Institute (GTI), Mohammed VI Polytechnic University, Ben Guerir 43150, Morocco
[3]LAMIH, CNRS, UMR 8201, Université Polytechnique Hauts-de-France, 59313 Valenciennes, France
[4]College of Engineering, University of Sharjah, Sharjah, United Arab Emirates
[5]Systèmes Industriels et Logistiques (SIL), University of Clermont Ferrand, 63170 Aubière, France

Corresponding author: Sara Mallah (saramallah.elec@gmail.com)

ABSTRACT In bulk port supply chains, products move from several source points (storage sheds) to various destination points (other storage sheds or vessels and trucks' loading stations). The concern in such ports is increasing the value of delivering customer-specific products on time by choosing the best transportation routes among a complex real-world belt-conveyor routes network. This routing problem plays a crucial role in reducing charges related to waiting and tardiness in loading and stocking operations. In this paper, we propose a mixed integer linear program that considers jointly: routing constraints and interactions with stocking and loading operations. We propose a second model with pre-processing and reduction techniques to enhance computational performance. The two models are tested on 126 generated instances with up to 1590 routes (the real matrix of routes of a real industrial application). The results reveal a striking difference between the first and the second models in terms of (*i*) the number of instances that could be handled (6 versus 111 out of 126); (*ii*) the number of instances solved to optimality (5% versus 71%). The models are promising and respond to the needs regarding integration between the routing, which is the main operation in the fertilizer and phosphate port supply chain, and other operations, such as stocking and loading. The integration of the proposed models toward a real-time planning/control integrated system is discussed.

INDEX TERMS Complex transportation routes' matrix, destination points, problems' interaction constraints, routing constraints, routing problem, source points.

I. INTRODUCTION

Maritime transport is a vital and indispensable sector for the continued distribution of critical supplies and international trade. Its importance was underscored in times of crisis [1], with the benefit accruing solely to more resilient markets. This redirection in maritime flows, for both container and bulk ports, resulted in a volume drop, albeit the nascent recovery in the second half of 2020.

To ensure resilience and efficiency in port operations, it is crucial to employ a decision-making tool that can handle complex optimizations regarding selecting the most suitable supply chain configurations. These appropriate configurations should enable serving clients on time. Hence, bottlenecks need to be avoided. Transporting commodities using a real-world conveyor-routes network can easily create multiple bottlenecks if proper decisions are not made. Thus, selecting appropriate routes for ordered products that meet industrial-setting constraints can help avoid penalties and increase total customer service. However, it should be noted that these industrial world constraints constitute a significant challenge to computational performance. This is because these industrial world constraints encompass constraints related to different bulk port problems, mainly: the

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

routing, loading, and storage spaces problems. The third problem comprises stocking and scraping problems but can be seen as one problem as they are carried out inside a storage shed [2].

The challenge is that these planning operations' problems must coordinate and integrate decisions to converge towards objectives that positively impact the bottom line. The opposite case would result in conflicting goals leading to delivering contaminated products on time or the right products with delays. Therefore, the integration of planning decisions of all operations making up an integrated optimization plan without contradictory goals, is vital for the efficient use of these bulk port operations. However, each of the problems is an NP-hard problem [3], [4], [5]. Consequently, the NP-hardness resulting from integrating them into one model would be significant. Hence simultaneous optimization-based integration (modelling all problems in one mathematical model) is irrelevant. Thus, the importance of an alternative integration approach would have problems being solved in separate models while interacting with each other rather than having one mathematical model to optimize them at once [6]. In other words, we would consider the constraints of each problem along with the constraints that arise from the interaction with other models. In this paper, we focus on the routing model while also considering the constraints that arise from the interaction with other bulk port problems. The reader can refer to our previous work[1] [6] to understand how the models' results are then consolidated to make up an integrated optimization plan. A preview of this integration approach is also described and schematized in Section III. This alternative approach is way more feasible regarding computational performance than if all problems are optimized simultaneously in one mathematical model, i.e., in the classical approach.

Furthermore, the alternative integration approach is relevant for its practical application in the industrial world, especially if real aspects need to be modeled as asked by the industrial interlocutor in our case study. On the contrary, due to the complexity of the simultaneous optimization-based integration, certain real aspects related to each problem are neglected in the classical approach, as will be demonstrated in Section II. Thus, in this paper, developing a routing problem with real routing constraints synchronized with real constraints of other problems' interaction is a first step towards successfully integrating the problems. We demonstrate in section IV that the routing model results promise to succeed in this integration approach.

As a case study, we consider the Jorf Lasfar bulk port, a Moroccan bulk cargo terminal of the group OCP known for its second rank worldwide in fertilizers and phosphate export. The OCP interlocutor confirmed that the routing is the major problem in this port and needs to be managed to handle two types of demands: loading demands, i.e., customers' demands, and stocking demands, i.e., internal demands to OCP when moving products from a storage shed to another storage shed is needed.

The loading demand can be composed of a set of products. Each product is to be served by a loader that constitutes the last equipment of a route. This loader refers to a gantry crane (pouring conveyor) when loading a vessel (a truck) (**Fig. 1**). Each destination point is directly connected to a series of successive conveyors back to the first conveyor into which the product was a priori poured by a colossal engine called scraper and located at the storage shed (See **Fig. 5**). In other words, the mission of the latter is to pick the product from the product stockpile and pour it into the conveyor with which it is directly connected. Then the product gets moved from one conveyor to another one until reaching the corresponding loader. For the stocking demand, the destination point is the stacker of the receiver storage shed (See **Fig. 5**). For both loading and stocking demands, each route can transport a product at a time from the source point (the receiving conveyor that is directly connected to the scraper) through a series of conveyors until reaching the corresponding destination point: a loader that loads a vessel's hold or a truck in case of a loading demand, or a stacker of the receiving storage shed in case of a stocking demand.

In this paper, the problem is to define (*i*) the sequence of transportation service of (vessels and/or trucks) loading demands and/or stocking demands, (*ii*) the corresponding route, and (*iii*) the transportation service time to transport each product of each demand. In the remainder, loading and stocking operations can be referred to as handling operations. The decisions to assign routes to demands are taken in such a way as to guarantee the respect of the stocking and loading plans while reducing the sum of tardiness penalties and waiting costs that might be incurred by the routing of the whole set of demands, given:

- The set of vessels' loading, trucks loading, and stocking demands,
- The precedence between demands that must be respected,
- The large number of routes with overlap nodes,
- The parallel or successive handling operations,
- The different capacities of equipment and routes (shown in **Fig. 1**),
- The time interval between successive products of the same demand,
- The pre-handling and post-handling durations of each demand that need to be respected for the safe start and finish of each handling operation,
- The presence of non-availability periods for vessels.

With such challenging constraints, one should propose a means to lighten the calculation of the model for reasonable computational performance. The solution to this issue is given and detailed in later sections of this paper, whose structure is as follows. State of the art is described in section II.

[1] When we mention 'our previous work,' it indicates that at least one author from the current paper was also an author of the cited work. The entire author list may not overlap between papers.

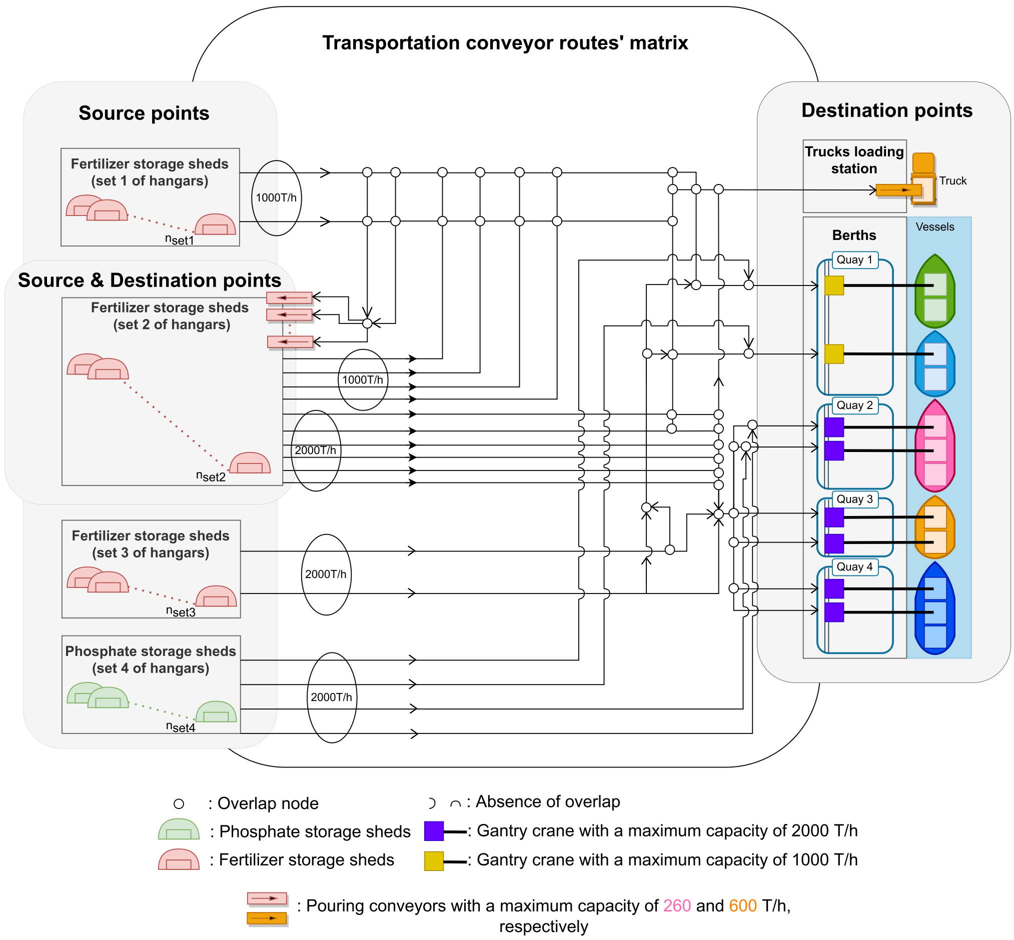


FIGURE 1. Moving products from sources to destination points via a conveyor routes' matrix in OCP Jorf Lasfar bulk port.

Section III describes the mathematical contributions to the routing problem. The computational results are analyzed in section IV. Section V discusses the managerial insights and the integration towards a real-time planning and control system. Finally, conclusions and perspectives are provided in section VI.

II. LITERATURE REVIEW

A. DISTINGUISHING THE BULK PORT BELT-CONVEYOR ROUTING PROBLEM FROM OTHER ROUTING PROBLEM TYPES

The optimization of bulk port belt-conveyor routing is crucial for the efficient management of bulk port supply chains. The slightest mistake in managing the conveyor transportation operation would prevent serving customers on time. Therefore, efforts should be deployed to model and optimize the routing problem harmoniously with other operations plans. However, in the literature, the routing problem in bulk ports has not gained enough attention in comparison to other problems, such as automated guided or load-dependent vehicle routing in container ports [7], [8], [9], [10], vehicle routing in multidoor cross-docking terminals [11], [12], vehicle routing in a general case [13], [14], [15], [16], [17], [18], and in case of picker routing problem [19], [20]. In most of these mentioned contributions, the optimal routes are selected depending on a routes-related parameter that can refer to the travel distance/time between the source and the destination and does not depend on parameters of other operations. Recently, Guo et al. [21] proposed a novel way to shorten the truck travel distance. However, this is possible within the context of the container terminal. They solved the coordinated optimization on berth allocation and yard assignment problem to obtain the optimal berthing and export container stacking positions in such a way as to minimize the total truck travel distance.

Nevertheless, in the routing of our research, which lies in a bulk port context, the product flow runs through conveyor routes connected to storage sheds and loaders. Thus, whatever the decision on loading and stocking problems, the travel distance cannot change and is always equal to the conveyor-route distance. Furthermore, the choice of optimal routes depends not only on the travel time of each route but also on the handling time (loading service time or stocking service time). Therefore, the bulk routing problem depends on the decisions of other problems as opposed to the references [7], [8], [9], [10], [13], [19], [20], [21], where only the routes-related parameters are considered.

B. COMPARING THE BULK PORT BELT-CONVEYOR ROUTING PROBLEM WITH CLASSICAL OPTIMIZATION PROBLEMS

The multi-commodity flow problem (MCFP) [22], [23], on the other hand, is a classical optimization problem that resembles the bulk port belt-conveyor routing problem in that it deals with the transportation of multiple commodities between various source and destination nodes in a network, except that the bulk port belt-conveyor routing problem is more constrained with bulk port context and its dependency on other bulk port problems, which makes it more complex. The MCFP is a generalization of the minimum cost flow problem [24], where there is only one source and one destination. The problem involves finding the optimal flow of multiple commodities through a network with capacity constraints. The multi-commodity flow problem is NP-hard, and therefore different methods were proposed to solve it, including mixed integer programming, heuristics, and metaheuristics [22], [23], [25].

C. THE IMPORTANCE OF INTEGRATING THE ROUTING PROBLEM WITH OTHER BULK PORT PROBLEMS AND THE DRAWBACKS OF SIMULTANEOUS OPTIMIZATION-BASED INTEGRATION

In this study, we are interested in modeling the belt-conveyor routing problem in such a way as to comply with the other operations' planning decisions without having to optimize the problems simultaneously in one mathematical model but instead by including constraints of interaction with the other bulk port optimization problems. However, to our knowledge, no work in the literature tackled the routing problem as we did. In the literature, the routing problem is viewed as a part of the whole integrated model, primarily through the work [26] in which Menezes et al. proposed a simultaneous optimization of stocking-related decisions with routing-related decisions in a bulk port for loading iron ore. However, certain critical decisions related to these operations are not covered, such as (*i*) whether to adopt a horizontal or vertical scraping strategy [27], as each strategy optimizes one aspect at the expense of the other. Specifically, the horizontal strategy maximizes the loaded volume, while the vertical strategy maximizes the storage space. (*ii*) the parallel or successive configuration of qualities' stocking or loading into ships was not covered either. (*iii*) The reduction of route complexity was achieved by classifying routes into subsets. However, this method is less advanced than the complexity reduction proposed in our paper (see section III-F). (*iv*) a sophisticated routes matrix was not considered (150 routes as a maximum). The same (apart from (*iii*)) applies to [28], in which Ago et al. also dealt with the simultaneous optimization of routing and stock problems of a bulk port for raw materials unloading for steel-making plants. Besides, in (*ii*), the loading is replaced with unloading, which can also be parallel or successive, but Ago et al. [28] did not study it.

Thus far, simultaneous optimization-based integration does not allow modeling all critical aspects of the problems. However, what is more noticeable is that, to our knowledge, the scientific literature has only handled the integration of a maximum of two problems in one mathematical model. For instance, both works of the last two mentioned references on bulk ports [26], [28]. In addition, Robenek et al. [29] modeled the integrated berth allocation and yard assignment problem in bulk ports. Unloading and stacking operations were optimized in an integrated fashion by Pratap et al. [30] as well. Unsal and Oguz [31] added to the scraper scheduling the stockyard allocation to make up one problem in addition to the second problem: berth allocation, as one subproblem of the loading problem without including the routing problem. In contrast, a scraper scheduling was proposed in [32] without considering the two scraping strategies, horizontal and vertical. et al. [33] were interested in the integration problem of unloading and storage space operations' decisions, namely the scraping and stocking. Nevertheless, the routing problem needs to be included in optimization. The work [34] is around integrating bulk ports' loading operation subproblems (multi-quays berth allocation and crane assignment) without considering the integration of routing and stocking problems. In most of these works, a further method to a mixed-integer programming model was needed. For instance, in the last cited work, the authors proposed some variants of the Variable Neighborhood Search (VNS) metaheuristic in order to tackle large-scale data sets.

All previous research contributions are relevant and significantly impacted the bulk port planning state of the art. However, in most of these contributions, the constraints of interaction between the problems in non-integrated problems are not included, and real-world constraints in integrated problems still need to be addressed. Nevertheless, most of the proposed models result in a large number of decision variables and constraints that complicate computation. Moreover, even if these models could converge in a reasonable time, they would optimize one or two operations at the expense of the third one not included in optimization. Furthermore, each bulk port problem has its own unique scope. For example, if a problem were to make a decision related to a second problem, it would do so at the expense of another constraint related to the second problem but not apparent to the first problem. As a result, the problem would be partial in its scope and that of the second problem, as previously commented on these contributions.

D. BENEFITS OF THE CHOSEN INTEGRATION APPROACH

In contrast to these works about simultaneous optimization-based integration, the coupling strategy between bulk port problems adopted in this paper is the alternative integration approach mentioned earlier in the introduction section and which was proposed in a former work [6], implies optimizing each problem separately, with each one having interaction constraints of the other problems. To broadly describe this coupling strategy, the different optimization problems must align their decisions with the port's objective of maximizing revenue and minimizing costs. This is why, among other reasons, the loading optimization subsystem is the first to be launched to initiate the decision-making process of the coupling strategy. In **Fig. 2** of Section III, we illustrate this coupling strategy by considering that the integrated optimization system is composed of optimization subsystems:

1) The loading optimization subsystem generates loading plans with maximum loaded volume and minimum costs and penalties respecting commercial plans (detailed documents outlining a company's sales and customers' demands and preferences for a specific period of time given at input). In the resulting loading plans, loading dates are provided with margins instead of rigid dates, thereby ensuring more feasibility to the subsequent optimization subsystem, whose mission must respect the loading plans.
2) The stocking optimization subsystem generates plans that respect loading plans and adjusts the latter if a stocking decision causes a loading date to shift. The shift would be carried out in the stocking optimization subsystem rather than in the loading optimization subsystem avoiding thereby any infinite loop with the loading optimization subsystem.
3) The routing optimization subsystem receives the loading and stocking plans with margins. It ensures that the plans are respected by generating via its first component (the routing model) (*i*) the optimal paths between the source and the destination points, (*ii*) the transportation windows respecting stocking and loading plans. The routing subsystem adjusts the loading and stocking plans with margins given as input using the second component (the adjustment block). As previously mentioned, the concept of margins is vital to confer flexibility on the returned plans and increase feasibility from one subsystem to the next.
4) The production optimization subsystem receives the resulting loading/stocking/routing plans to ensure the production on time of the requested products while meeting the time windows of routing, stocking, and loading plans.

Overall, this coupling strategy will allow the routing model to have, automatically, some constraints relaxed, given that the decisions specific to stocking and loading are made within the stocking and loading models, respectively. For instance, the stocking decision concerning the choice of the storage shed (the source point) from which an ordered product will be supplied to a given customer. In addition to the loading decision, which pertains to the specification of the gantry crane (i.e., the destination point) that will supply the customer with the product. Hence, the routing problem focuses on the complexity of its perimeter, mainly in terms of a sophisticated, real-world conveyor routes matrix, while considering the interaction constraints and parameters that would allow the routing problem to synchronize with the other problems for a smooth and successful service.

E. AN ANALYSIS OF SIMILARITIES AND DIFFERENCES BETWEEN THE CHOSEN INTEGRATION APPROACH AND A SIMILAR APPROACH FROM THE LITERATURE

Conversely, the literature is rich in a roughly similar approach, commonly known as the bi-level optimization approach. The two approaches are similar since the latter is also used when dealing with problems that involve multiple interconnected decision-making levels [35], with one decision-maker (upper-level problem) influencing the decisions of another decision-maker (the lower-level problem). The lower-level problem's optimal decisions, in turn, affect the upper-level problem [36], [37]. The only difference that makes our integration approach more favorable regarding computational performance, especially within the industrial context, is manifested in how the lower-level problem's optimal decisions affect the upper-level problem. In the bi-level optimization approach, the optimal solution of the lower-level problem feeds back into the upper-level problem so that the latter can adjust its decision for another cycle of the hierarchical decision-making process. Indeed, the relationship between the upper-level and the lower-level problems is hierarchical. The former is called the leader, and the latter is called the follower, making the approach computationally expensive to solve, especially when the lower-level problem is complex and needs to be solved repeatedly [36]. In other words, a typical scenario of a bi-level optimization problem may be described as the following (1) the upper-level problem (the leader) generates an optimal solution, (2) the lower-level problem (the follower) reacts to the returned optimal solution by generating, in turn, an optimal solution that respects the optimal solution of the leader, (3) the optimal solution generated by the lower-level problem at (2) feeds back into the upper-level problem so that the latter can evaluate it (4) The leader adjusts its decisions based on the feedback from the follower's response. (5) Steps 2-4 are repeated until an equilibrium is reached, where no solution in the decision space of the follower is available. However, in our integration approach, there is no such hierarchical relationship between the problems. As the coupling strategy was described, the lower-level problem generates solutions (plans) that respect the upper-level problem's solutions (plans) and adjusts the latter if a decision of the lower-level problem causes the date of the solution of the upper-level problem to shift. The shift would be carried out in the lower-level problem rather than in the upper-level problem avoiding thereby any infinite loop with the upper-level problem. Still, the bi-level optimization approach resembles our adopted integration approach in many ways, namely (i) allowing a clear separation of decision levels by clearly distinguishing between the leader's and the follower's objectives and constraints and by capturing the interdependencies between the decisions, allowing for more structured analysis and understanding of the interactions between the two decision-making levels. This structure makes it suitable for modeling complex real-world problems where decisions are made at different levels leading thereby to better solutions [35], which enables realistic modeling as it allows for the natural modeling of situations where one decision-maker (the leader) affects the decision space of another decision-maker (the follower); (ii) allowing flexibility by being a framework that can accommodate various types of decision-making problems and objectives, making it adaptable to various application areas, including transportation, energy, and supply chain management; (iii) being challenging to solve, particularly when dealing with combinatorial objectives and constraints due to the interdependence between the leader's and the follower's decisions. Advanced solution techniques, such as metaheuristics, decomposition methods, or specialized exact algorithms, may be required to tackle these problems [38] efficiently; (iv) being difficult to apply to real-world problems, as it requires a clear understanding of the decision-making process and the relationships between the upper and lower levels of decision-making [37]. Overall, the bi-level optimization approach is a powerful tool for solving complex real-life optimization problems that involve multiple levels of decision-making, such as transportation planning, supply chain management to optimize both strategic and tactical decisions in the supply chain network design [39], [40], and energy systems [41]. In transportation planning, for example, the upper-level decision-maker may decide on the number of buses to purchase, while the lower-level decision-maker may decide on the optimal bus routes [38]. However, in terms of solution techniques, developing efficient and effective solution techniques for bi-level optimization problems remains an ongoing research challenge. While some specialized exact algorithms and decomposition methods have been proposed, many bi-level problems require the use of heuristics or metaheuristics to find good solutions within reasonable computational times for large-scale data sets [42], [43], [44], [45], [46]. Finally, solving the bulk port problems using the bi-level optimization approach is legitimate. Still, whether we opt for it or for the integration approach we proposed, the lower-level problem remains crucial. This problem, specifically the bulk port belt-conveyor routing issue, represents the main contribution of our paper, and it should be addressed in such a way as to minimize changes to the decisions of the upper-level problem.

Compared with the literature, the study of this paper is motivated by the alternative integration approach to overcome the shortcomings of the classical integration approach, "simultaneous optimization-based integration", and the shortcoming of the "bi-level" optimization approach. Our study is particularly interested in developing the bulk port belt-conveyor routing model (as architected in **Fig. 2**) while including real-world constraints of conveyor routes transportation and while considering interaction constraints with the other problems, namely loading and stocking problems. The routing model established aims to minimize the sum of waiting costs and tardiness penalties according to loading and stocking demands or plans, which include dates with margins. Owing to the challenge the constraints above can make, reducing the complexity of routes' configuration is addressed to improve the computational performance. The way of doing so is compared to the literature in Section III.

III. PROBLEM FORMULATION

Fig. 1 shows the combinatorial problem of routing conveyor paths, which involves various itineraries with different capacities from three sets of fertilizer storage sheds and one set of phosphate sheds, connecting to four quays with two gantry cranes each, as well as a truck loading station with a pouring conveyor as a loader. Additionally, transportation is required to move fertilizer products from storage sheds in set 1 to those in set 2. The phosphate and fertilizers products both comprise different qualities.

The routing model under study receives the planning of demands from the loading and stocking subsystems, as detailed in **Fig. 2**. These plans consist of stocking and loading decisions, such as specifying the source and destination points, respectively. The loading and stocking constraints, which are expressed in terms of interaction problems-based parameters, work in harmony with the routing constraints expressed in terms of routing parameters. This synchronization ensures that transportation operations align with loading and stocking operations, as required in industrial scenarios. The notation for the general routing parameters is provided in Section III-C1, while the notation for the interaction problems-based parameters is explained in Section III-C2. However, before delving into the notation, it is worth providing more details about some of these parameters, particularly the configuration of qualities (see Section III-A) and time parameters (see Section III-B).

The scope of this paper covers the routing model, whose mission is to program the transportation of loading and stocking demands in such a way as to minimize the total waiting costs and tardiness penalties according to loading and stocking dates. Updating the latter after the routing model decisions is to be carried out by another block of the routing optimization subsystem, as was discussed in **Fig. 2**. Only then, can the production optimization subsystem handle loading, stocking, and routing demands.

A. QUALITIES-BASED SPECIFICATIONS

Each demand is characterized by a list of qualities that should be transported to a destination. These qualities are set to be handled (loaded or stocked) in a specific configuration. The latter is parallel if the number of handling equipment is more than one. It is successive if the number of qualities to be handled by each handling equipment is greater than or equal to one. There can be a mixture of the two configurations to end up having as many lists of successive qualities as there are handling equipment. In any case, the total number of qualities cannot exceed the number of the existing container spaces, be they holds of a vessel, truck container, or storage

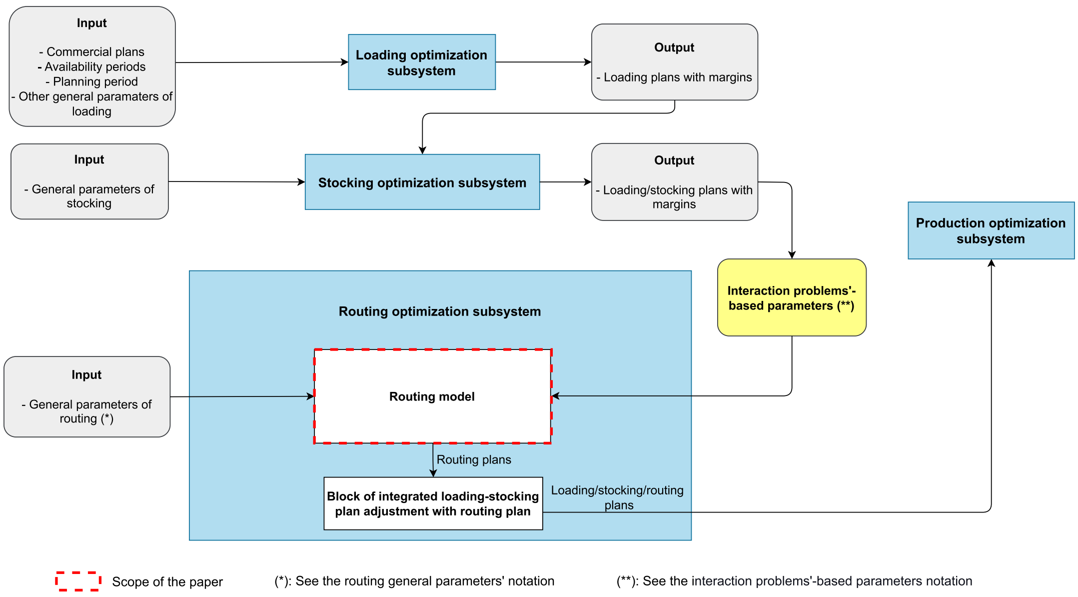


FIGURE 2. Interactions between integrated bulk port problems' models.

spaces of a storage shed. Some examples of these demands' configurations are illustrated in **Fig. 3** and **Fig. 4**.

Fig. 3 illustrates two vessels' demands. The first demand is served by only one handling equipment, a gantry crane to load successively three qualities in the three holds existing for vessel 1. For the second demand, two gantry cranes load two successive lists of qualities in parallel. The first successive list includes two qualities q_1 and q_2. The second list includes one quality q_3. Therefore, q_1 and q_3 are loaded at the same time, while q_2 is to be loaded after the end of loading q_1 and q_3.

Fig. 4 illustrates a truck demand (demand 3) and a stocking demand (demand 4). The truck is endowed with only one container. So, it can be loaded with only one product by only one pouring conveyor at a time. Conversely, in demand 4, a storage shed can have a couple of storage spaces. However, it is equipped with only one stacker. So, the parallel configuration is not possible for the stocking demands either, but the number of successive qualities can be greater than or equal to one. There can be as many successive qualities as there are vacant storage spaces in a storage shed.

B. TIME PARAMETERS

1) TRANSFER, HANDLING AND TRANSPORT TIMES

The three routes represented in **Fig. 5** are for stocking, truck loading, and vessel loading demands. For each of these demands, a corresponding conveyors' route moves a quality from the source (scraper) to the destination (stacker, gantry crane, or pouring conveyor of the truck station). The amount of time needed for the first seed of the product to travel this distance from the source to the destination is the travel time or the transfer time. At precisely the end of the latter, can the handling operation (stocking or loading) begin. The handling operation ends if and only if we finish (stocking or loading) the total quantity requested. Therefore, the transport time that consists of transporting the whole quantity of a quality is the sum of the transfer time and the handling time. **Fig. 5** illustrates this formula.

2) EARLIEST AND LATEST DATES

Due to the concept of margins mentioned earlier, the optimization stocking and loading subsystems generate loading-stocking plans with time windows rather than rigid dates to give the routing model flexibility upon transport's start and end dates. This implies time parameters such as earliest start date, earliest and latest end dates.

- Earliest end date = earliest start date + handling time
- Latest end date = earliest end date + total margins' duration
- Time window = [earliest start date; latest end date]

To better understand these time parameters, the reader might refer to **Fig. 6**. It illustrates these time parameters for vessels and trucks' loading as examples of handling operations.

For the routing problem, rather than handling (stocking/loading) parameters, we are interested in transport time parameters which can be deducted from the formula that links transport time and handling time. The handling time is generated by the loading/stocking suboptimizers, while the transfer time is a fixed parameter related to each route.

Further time parameters pinpointed in **Fig. 6** are the pre-handling and post-handling durations. They are necessary to perform port-related tasks before and after a handling service.

The precedence notion is paramount for demands to which the same resource (space or handling equipment) is assigned during overlapping time periods. It sets the predecessor demand of a demand and the time interval between them. For

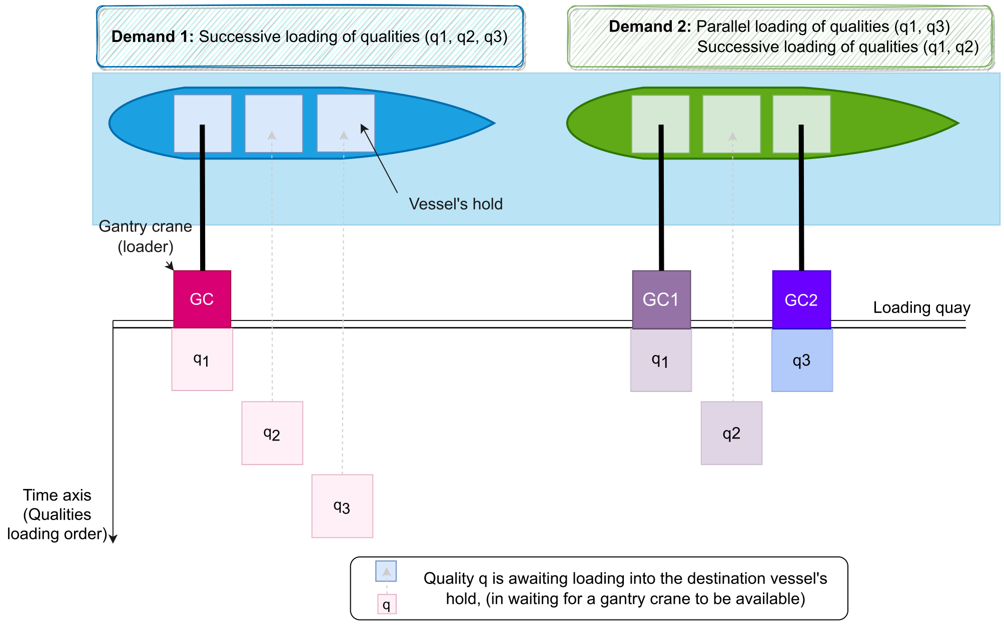


FIGURE 3. Products (qualities') configurations in vessels' demands.

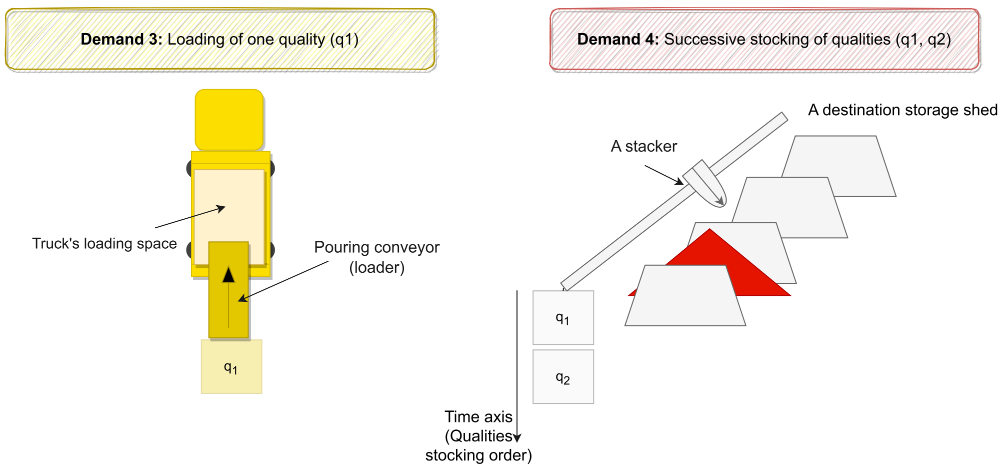


FIGURE 4. Qualities' configurations in stocks and trucks' demands.

instance, in **Fig. 6**, vessels 1 and 2 are allocated to be served at the same quay space using a common gantry crane. The same thing applies to vessels 1 and 3.

3) WAITING TIME AND TARDINESS

Despite the margins-related flexibility to start and end whenever possible within the transport time window, the ultimate decision for the routing problem consists of selecting, for demands, routes with shorter transport time, minimizing the sum of tardiness penalties and waiting costs regarding transport, loading and stocking time-windows and constraints. Given that:

- The tardiness of a service is the positive value of its real end date minus its expected end date [47].
  Hence:
  Tardiness of transport = max(0, real end date of transport - latest end date of transport) = max(0, real start date of transport + real transfer time + handling time - latest end date of transport)
- The waiting time of a service is the positive value of its real start date minus its expected start date.
  Hence:
  Waiting time of transport = max(0, real start date of transport - earliest start date of transport).

The model is proposed considering the following hypotheses:

- The transport of a demand must be continuous, i.e., no suspension is authorized between the transport of

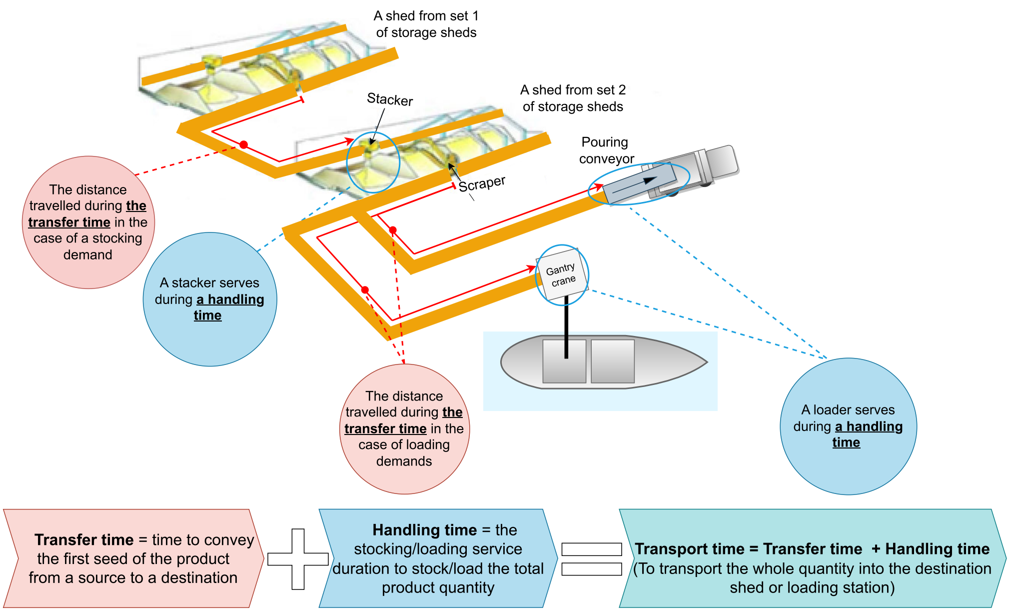


FIGURE 5. Transfer, handling and transport times.

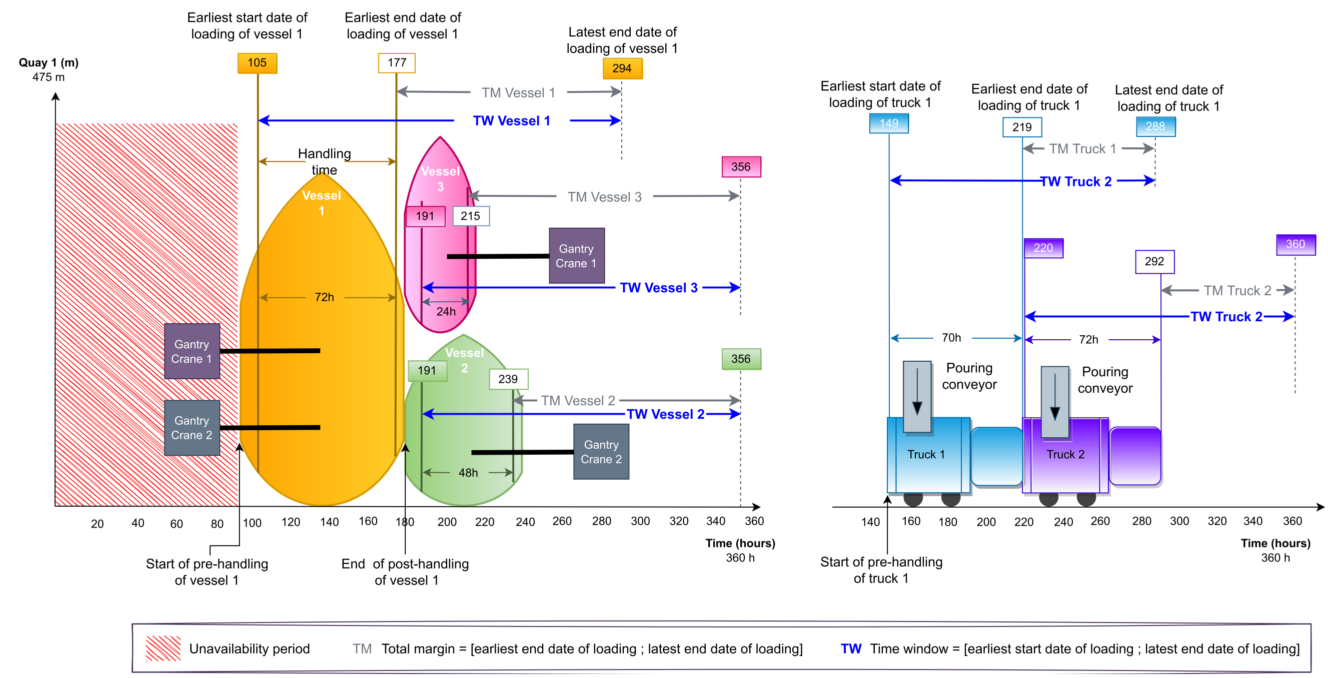


FIGURE 6. Margins-related time parameters.

the qualities requested, except the time needed between successive ones.

- The maximum number of handling equipment used to satisfy a demand at a time is two
- The number of successive qualities served by the first handling equipment is greater than or equal to the number of successive qualities served by the second handling equipment.
- The real transfer time is less than or equal to 1.

C. NOTATION

1) ROUTING GENERAL PARAMETERS' NOTATION

R	The set of routes
E	The set of equipment
R_e	The set of routes that share the same equipment, such as $R_e \subseteq R$
tt_r	The real transfer-time
tt_e	The expected transfer-time
C_r	The capacity (in tons/hour) of each route $r \in R$

e_{er} — 1 if an equipment $e \in E$ belongs to the route r, 0 otherwise

2) INTERACTION PROBLEMS'-BASED PARAMETERS NOTATION

D — The set of demands

γ_d — The number of handling equipment used to satisfy the demand d ($d \in D$)

Q_d — $= Q_d^1 \cup Q_d^2$. Set of sets of successive qualities. If $\gamma_d = 1$, Q_d is composed of one set of successive qualities of the demand d. $Q_d^1 = \{1, \ldots, nQ_d^1\}$, $Q_d^2 = \varnothing$. If $\gamma_d = 2$, Q_d is composed of two sets of successive qualities of the demand d. $Q_d^1 = \{1, \ldots, nQ_d^1\}$, $Q_d^2 = \{nQ_d^1 + 1, \ldots, nQ_d^1 + nQ_d^2\}$

P — A set of demand pairs $(d, d') \in P$

$\Delta_{dd'}$ — The time needed between transporting the demands d and d'

s_d — The earliest start date of the transportation service of the demand d

f_d — The latest end time of the transportation service of the demand d

H — The time horizon $[0, H_{max} - 1]$

H_{max} — The total number of time steps in the time horizon

H_d — $= [s_d, H_{max} - 1]$. Transportation horizon of the demand d

h_{dq} — The handling time to satisfy the quality q of the demand d

b_{dq} — The extra period that the parallel quality, to q of a demand d, needs to end its handling if and only if $\gamma_d = 2$, 0 otherwise

δ — The time needed between the transport of two successive qualities of the same demand

o_{dq} — The source point of the quality q of the demand d

g_{dq} — The destination point of the quality q of the demand d

α_d — Tardiness penalty for the demand d

β_d — Waiting cost for the demand d

l_{dq} — The handling flowrate (in tons/hour) of the quality q of the demand d

A_d — Set of available periods for the demand d

a_p^{ds} — The start date of the available period $p \in A_d$

a_p^{df} — The end date of the available period $p \in A_d$

pre_d — The pre-handling duration of the demand d

$post_d$ — The post-handling duration of the demand d

D. VARIABLES

T_d — Tardiness of the demand d

W_d — Waiting time of the demand d

x_{dqrt} — Equals to 1 if the route r starts transporting the quality q of the demand d at the period t and equals to 0 otherwise

y_{dp} — Equals to 1 if the demand d is transported during the available period $p \in A_d$

E. MATHEMATICAL MODEL: R-ROUTING

$$\min \quad \sum_{d \in D} \alpha_d \cdot T_d + \beta_d \cdot W_d \tag{1}$$

$$\sum_{r \in R} \sum_{t \in H_d} t \cdot x_{dqrt} - s_d + H_{max} \cdot \left(1 - \sum_{r \in R} \sum_{t \in H_d} x_{dqrt}\right) \leq W_d \quad \forall d \in D, q = 1 \tag{2}$$

$$\sum_{r \in R} \sum_{t \in H_d} (t + tt_r + h_{dq_i}) \cdot x_{dq_irt} - f_d + \left(1 - \sum_{r \in R} \sum_{t \in H_d} x_{dq_irt}\right) \cdot \left(H_{max} + \delta \cdot (nQ_d^i - 1) + \sum_{k=q_i'}^{q_i} tt_e + h_{dk}\right) \leq T_d$$
$$\forall d \in D, i = 1 \text{ if } \gamma_d = 1\,;\ i \in \{1, 2\} \text{ if } \gamma_d = 2,$$
$$(q_i', q_i) = \begin{cases} (1, nQ_d^1), & i = 1 \\ (nQ_d^1 + 1, nQ_d^1 + nQ_d^2), & i = 2 \end{cases} \tag{3}$$

$$\sum_{r \in R} \sum_{t \in H_d} x_{dqrt} \leq 1 \qquad \forall d \in D, \quad \forall q \in Q_d \tag{4}$$

$$\sum_{d \in D} \sum_{q \in Q_d} \sum_{r \in R} \sum_{t=0}^{s_d - 1} x_{dqrt} = 0 \tag{5}$$

$$\sum_{r \in R} \sum_{t \in H_d} (t + \lceil tt_r \rceil + h_{dq} + \delta + b_{dq}) \cdot x_{dqrt} = \sum_{r \in R} \sum_{t \in H_d} t \cdot x_{dq'rt} \quad \forall d \in D,$$
$$\forall q \in Q_d \backslash \{nQ_d^1\} \text{ if } \gamma_d = 1,$$
$$\forall q \in Q_d \backslash \{nQ_d^1, nQ_d^1 + nQ_d^2\} \text{ if } \gamma_d = 2,$$
$$q' = q + 1 \tag{6}$$

$$\sum_{r \in R} \sum_{t \in H_d} \left(t + \lceil tt_r \rceil\right) \cdot x_{dqrt} \quad \forall d \in D, \gamma_d = 2,$$
$$= \sum_{r \in R} \sum_{t \in H_d} \left(t + \lceil tt_r \rceil\right) \cdot x_{dq'rt} q = 1,$$
$$q' = nQ_d^1 + 1 \tag{7}$$

$$\sum_{r \in R} \sum_{t \in H_d} (t + tt_r + h_{dq} + b_{dq} + \Delta_{dd'}) \cdot x_{dqrt} \leq \sum_{r \in R} \sum_{t \in H_{d'}} t \cdot x_{d'q'rt} + H_{max} \cdot \left(1 - \sum_{r \in R} \sum_{t \in H_{d'}} x_{d'q'rt}\right) \quad \forall (d, d') \in P,$$
$$q = nQ_d^1, q' = 1 \tag{8}$$

$$\sum_{r \in R} \sum_{t \in H_d} l_{dq} \cdot x_{dqrt} \leq \sum_{r \in R} \sum_{t \in H_d} C_r \cdot x_{dqrt}$$

$$\forall d \in D, \quad \forall q \in Q_d \tag{9}$$

$$\sum_{d' \in D} \sum_{\substack{q' \in Q_{d'} \\ q' \neq q \text{ if } d'=d}} \sum_{r' \in R_e} \sum_{t'=\sigma}^{\rho} x_{d'q'r't'} \quad \forall d \in D,$$

$$\leq \left(\sum_{d \in D} | Q_d | \right) \cdot (1 - x_{dqrt}) \quad \forall q \in Q_d,$$

$$\forall e \in E, \quad \forall r \in R_e, \quad \forall t \in H_d,$$

$$\sigma = \lceil \max(0, t - h_{d'q'} - tt_{r'}) \rceil,$$

$$\rho = \lfloor \min(H_{max} - 1, t + tt_r + h_{dq}) \rfloor \tag{10}$$

$$\sum_{t \in H_d} x_{dqrt} \leq e_{o_{dq}r} \cdot e_{g_{dq}r} \quad \forall d \in D,$$

$$\forall q \in Q_d, \quad \forall r \in R \tag{11}$$

$$\sum_{p \in A_d} y_{dp} \leq 1 \quad \forall d \in D \tag{12}$$

$$\sum_{p \in A_d} (a_p^{ds} + pre_d - \lceil tt_e \rceil) \cdot y_{dp} \qquad \forall d \in D,$$

$$\leq \sum_{r \in R} \sum_{t \in H_d} t \cdot x_{dqrt} q = 1 \tag{13}$$

$$\sum_{r \in R} \sum_{t \in H_d} (t + tt_r + h_{dq} + b_{dq}) \cdot x_{dqrt}$$

$$\leq \sum_{p \in A_d} (a_p^{df} - post_d) \cdot y_{dp} \qquad \forall d \in D, q = nQ_d^1 \tag{14}$$

$$T_d \geq 0 \quad \forall d \in D \tag{15}$$

$$W_d \geq 0 \quad \forall d \in D \tag{16}$$

$$x_{dqrt} \in \{0, 1\}$$

$$\forall d \in D, \quad \forall q \in Q_d, \quad \forall r \in R, \quad \forall t \in H \tag{17}$$

$$y_{dp} \in \{0, 1\}$$

$$\forall d \in D, \quad \forall p \in A_d \tag{18}$$

The objective function (1) seeks to minimize the total penalty tardiness and waiting costs. The waiting time and the tardiness of each demand d are calculated by the set of constraints (2) and (3), respectively. Constraints (4, 5) ensure that the transport operation of each demand quality starts at the most one time in the given horizon. In contrast, each demand can only be transported after the earliest start date of transport set for it. Constraints (6) establish the order in transporting the successive qualities of each demand with or without the parallel configuration. Whereas constraints (7) set the parallel condition that implies handling parallel qualities should start simultaneously. Constraints (8), on the other hand, set the order for each demand d′ and its predecessor d. Constraints (9) ensure that the chosen route has the capacity needed to transfer the requested quality. Constraints (10) avoid time overlap between overlapping routes with the same equipment. It is important to note that any equipment within a route, including the source, destination, or any equipment in between, can potentially be an overlap node if it is shared with another route or multiple routes. Constraints (11) set for each demand quality, routes with the source and destination points specified. Constraints (11) work with constraints (10) to avoid overlapping problems. The constraints (12-14) are used to schedule a demand within an available period considering the pre-handling and post-handling durations in which no operation should run. Constraints (15, 16) and (17, 18) set the continuous variables for each demand and the binary ones, respectively. In the model, we consider the whole set of routes R. Hence the name R-routing model.

In terms of big O, the R-routing model has $O(|D| \cdot |Q| \cdot |R| \cdot |H|)$ variables and $O(|D|^2 \cdot |Q|^2 \cdot |E| \cdot |R_{e\,\max}|^2 \cdot |H|)$ constraints, where $|D|$ is the number of demands, $|Q|$ is the maximum number of qualities for all demands, $|R|$ is the maximum number of routes R, $|H|$ is the maximum time periods for all demands, and $|R_{e\,\max}|$ is the maximum number of routes R_e for all equipment checked throughout the iterations.

F. MATHEMATICAL MODEL IMPROVEMENT: R_{dq}-ROUTING

The initial model proposed in this study aimed to select, for each quality of each demand, a route from the complete set of total routes R. This posed some computational challenges because the R-routing model had to select the most time-efficient route while also considering the adequacy of the route for a specific demand quality. However, in an effort to refine our model and enhance its computational performance, we developed an improvement. This improvement, implemented through Algorithm 2, curates specific routes for each quality q of a demand d: R_{dq}, a priori, leaving up the selection of the most time-efficient routes to the model. Consequently, the model now considers a subset of routes R_{dq} instead of the entire set R.

Taking this preprocessing work-based refinement a step further, we introduced R_{dqe}, representing the intersection of R_{dq} and R_e, which will be utilized in overlapping constraints. This change aims to reduce complexity during routing plan generation, potentially boosting the model's performance. Therefore, we are introducing new notation into the model, believing that these improvements will significantly enhance the model's efficiency and computational performance.

- R_{dq} A subset of R, contains all the routes that link the source from where the quality q of the demand d is transported to its destination. It also refers to the subset of routes with the same source and destination.
- R_{dqe} is a subset of routes obtained from the intersection of R_{dq} and R_e. The latter, as mentioned earlier, is the set of routes that share the same equipment e. $R_{dqe} = R_{dq} \cap R_e$

In order to obtain the set R_{dq}, we utilize the set of routes R, where each route is defined by its source o (for origin), destination g (for goal), and capacity C_r. The latter is already specified in notation. We then proceed to create a set S that contains, for each source o, a set R_o containing all the routes

Algorithm 1 Prepare S (set of Routes by source)

```
   Input: Set of all routes R
1  for r ∈ R do
2  |  o ← r.source
3  |  g ← r.destination
4  |  if o ∉ S then
5  |  |  R_o ← ∅
6  |  |  R_og ← ∅
7  |  |  R_og ← R_og ∪ {r}
8  |  |  R_o.append(g, R_og)
9  |  |  S.append(o, R_o)
10 |  else
11 |  |  if g ∉ R_o then
12 |  |  |  R_og ← ∅
13 |  |  |  R_og ← R_og ∪ {r}
14 |  |  |  R_o ← R_o ∪ R_og
15 |  |  else
16 |  |  |  R_og ← R_og ∪ {r}
17 |  |  end
18 |  end
19 end
20 return S
```

originating from that particular source o. The set R_o is further characterized by destination g. In fact, for each destination g, we have a set R_{og} comprising of all routes that share the same destination g and that originate from a particular source o.

To construct the set S, we propose Algorithm 1. The for loop iterates through all the routes in R, which takes $O(|R|)$ time. The subsequent steps, particularly the search operations carried out in line 4 and line 11 of the algorithm, each have a time complexity of $O(1)$. This is because these searches are performed within a C++ set, a data structure which indeed allows for constant time operations. Furthermore, given the non-nested structure of these two steps within the algorithm, their complexities are additive. Hence, in the worst-case scenario, the time complexity of the algorithm is accurately represented as Hence, in the worst case, the time complexity of the algorithm is $O(|R|)$.

After the set S is constructed, we proceed to prepare the set R_{dq} routes using Algorithm 2. The set R_{dq} is formed by routes that have a specific source o and destination g, and belong to a specific quality q of a demand d, and whose capacity C_r is the maximum of the handling flowrate l_{dq} of the demand quality in question. The time complexity of the algorithm is $O(|D| \cdot |Q| \cdot \alpha)$, where $|D|$ is the number of demands, $|Q|$ is the maximum number of qualities for all demands, and α is a value through which R_{og} can be upper bounded.

The R_{dq}-routing model is obtained by replacing, in the R-routing model, R with R_{dq} but also by considering two additional changes:

- In the constraints (10): $\forall r \in R_e$ and $r' \in R_e$ should be replaced by $\forall r \in R_{dqe}$ and $r' \in R_{d'q'e}$ respectively. Therefore, the constraints (10) in the R_{dq}-routing model

Algorithm 2 Prepare R_{dq} Routes

```
Input: Set of routes by source S, Set of all demands D
for d in D do
|  for q in d.qualities do
|  |  R_dq ← ∅
|  |  o ← o_dq, g ← g_dq
|  |  for r in S[o].R_o[g].R_og do
|  |  |  if l_dq ≤ C_r then
|  |  |  |  R_dq ← R_dq ∪ {r}
|  |  |  end
|  |  end
|  end
end
return R_dq
```

become:

$$\sum_{d' \in D} \sum_{\substack{q' \in Q_{d'} \\ q' \neq q \text{ if } d'=d}} \sum_{r' \in R_{d'q'e}} \sum_{t'=\sigma}^{\rho} x_{d'q'r't'} \leq \left(\sum_{d \in D} |\, Q_d \,| \right) \cdot (1 - x_{dqrt}) \qquad \forall d \in D,$$
$$\forall q \in Q_d, \forall e \in E, \forall r \in R_{dqe}, \forall t \in H_d,$$
$$\sigma = \lceil \max(0, t - h_{d'q'} - tt_{r'}) \rceil,$$
$$\rho = \lfloor \min(H_{max} - 1, t + tt_r + h_{dq}) \rfloor$$

- An additional set of constraints (19) for the R_{dq}-routing model:

$$\sum_{r \in R \setminus \{R_{dq}\}} \sum_{t=0}^{H} x_{dqrt} = 0 \quad \forall d \in D, \forall q \in Q_d \tag{19}$$

Constraints (19) prohibit routes that do not belong to R_{dq}.

In terms of big O, the R_{dq}-routing model has $O(|D||Q||R_{dq\,\max}||H|)$ variables and $O(|D|^2|Q|^2|E||R_{dqe\,\max}|^2 |H|)$ and constraints, where $|R_{dq\,\max}|$ is the maximum number of routes R_{dq} for all demands and qualities, and $|R_{dqe\,\max}|$ is the maximum number of routes $R_{dqe}|H|$ is the maximum number of routes R_{dqe} for all equipment checked throughout the iterations. Therefore, the difference between the R-routing model and the R_{dq}-routing model in terms of complexity lies in the number of routes checked through all iterations from $|R|$ to $|R_{dq\,\max}|$ and from $|R_{e\,\max}|$ to $|R_{dqe\,\max}|$.

To the best of our knowledge, only Menezes et al. [26] proposed a way to reduce the complexity of a transportation matrix of routes. They suggested an idea that we managed to try to see the difference with ours (R_{dq}). They propose dividing the whole set of routes into three subsets of routes. Where each one of them transports products from a sources' set to a destinations' set. Following this example, we ended up with nine subsets of routes as the numbers of sources' sets and destinations' sets, in our case, are more significant and as they have different maximal capacities. We denote R_k, with

TABLE 1. Comparison of R_k and R_{dq} as sub-sets of the whole routes' set R.

Set of routes	R	R_{dq}	R_k
Maximal number of routes	55	1	18
	105	2	24
	1590	32	760

$k = \{x, y, z', z'', v', v'', w, s, t\}$. **Table 1** shows that for each set of routes R, R_{dq} reduces the number of routes into fewer routes than R_k does with a sharp difference. This finding is rational as R_k depends on the type of routes independently of the demand. At the same time, R_{dq} is more specific and divides routes into subsets depending on each demand's quality.

The computational results of the R_k model, proposed in [26], cannot be compared with the R_{dq}/R-models as the former includes routing and stocking decisions in one model. In contrast, the latter models include routing decisions only.

G. A HYPOTHESIS RELAXATION

The contribution we propose in this section is in order to relax the hypothesis of having a real transfer time that is less than or equal to a time unit ($tt_r \leq 1$) and thus have a flexible model that represents all real-world bulk ports' cases, albeit most bulk ports, generally speaking, are constructed in such a way that the transfer time does not exceed 1 hour. While trying to relax this hypothesis, we had a quadratic expression. However, to convert the nonlinear resulted program into a linear program, we introduced two additional parameters: M, as a sufficiently large constant. In our case, M is the maximum transfer time. The second parameter is λ_d which is the difference between nQ_d^1 and nQ_d^2. In addition to the two following variables:

- v_{dq} Takes the maximum integer value from the transfer time to transport the quality q of the demand d and the transfer time to transfer the parallel quality to q of the same demand d.
- y_{dq} Equals 1 if the time to transfer the quality q of the demand d is less than the time needed to transfer its parallel quality of the same demand d; equals 0 otherwise.

Thus, the model's set of constraints (6) is replaced by constraints (20-26). The constraints (7) of the model are replaced by constraints (27). The latter constraints ensure the same loading start for every two parallel qualities.

$$\sum_{r\in R_{dq}}\sum_{t\in H_d}(t + \lceil tt_r\rceil + h_{dq} + \delta + b_{dq}) \cdot x_{dqrt} = \sum_{r\in R_{dq'}}\sum_{t\in H_d} t \cdot x_{dq'rt}$$
$$\forall d \in D, \forall q \in \{1, \ldots, nQ_d^1 - 1\} \text{ if } \gamma_d = 1,\ \forall q \in \{nQ_d^1 - \lambda_d, \ldots, nQ_d^1 - 1\} \text{ if } \lambda_d > 0 \text{ and } \gamma_d = 2, q' = q + 1 \tag{20}$$

$$\sum_{r\in R_{dq}}\sum_{t\in H_d}(t + \lceil tt_r\rceil + h_{dq} + \delta + b_{dq}) \cdot x_{dqrt} = \sum_{r\in R_{dq'}}\sum_{t\in H_d}(t + \lceil tt_r\rceil) \cdot x_{dq'rt} - v_{dq'}$$
$$\forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{1, \ldots, nQ_d^1 - \lambda_d - 1\}, q' = q + 1 \tag{21}$$

$$v_{dq} \geq \sum_{r\in R_{dq}}\sum_{t\in H_d}\lceil tt_r\rceil \cdot x_{dqrt}\quad \forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{2, \ldots, nQ_d^1 - \lambda_d\} \tag{22}$$

$$v_{dq} \geq \sum_{r\in R_{dq'}}\sum_{t\in H_d}\lceil tt_r\rceil \cdot x_{dq'rt}\quad \forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{2, \ldots, nQ_d^1 - \lambda_d\}, q' = q + nQ_d^1 \tag{23}$$

$$v_{dq} \leq \sum_{r\in R_{dq}}\sum_{t\in H_d}\lceil tt_r\rceil \cdot x_{dqrt} + (1 - y_{dq}) \cdot M\quad \forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{2, \ldots, nQ_d^1 - \lambda_d\} \tag{24}$$

$$v_{dq} \leq \sum_{r\in R_{dq'}}\sum_{t\in H_d}\lceil tt_r\rceil \cdot x_{dq'rt} + (1 - y_{dq'}) \cdot M\quad \forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{2, \ldots, nQ_d^1 - \lambda_d\}, q' = q + nQ_d^1 \tag{25}$$

$$y_{dq} + y_{dq'} = 1\quad \forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{2, \ldots, nQ_d^1 - \lambda_d\}, q' = q + nQ_d^1 \tag{26}$$

$$\sum_{r\in R_{dq}}\sum_{t\in H_d}(t + \lceil tt_r\rceil) \cdot x_{dqrt} = \sum_{r\in R_{dq'}}\sum_{t\in H_d}(t + \lceil tt_r\rceil) \cdot x_{dq'rt}\quad \forall d \in \{d \mid d \in D \text{ and } \gamma_d = 2\},\ \forall q \in \{2, \ldots, nQ_d^1 - \lambda_d\}, q' = q + nQ_d^1 \tag{27}$$

$$v_{dq} \geq 0 \quad \forall d \in D, \forall q \in Q_d \tag{28}$$

$$y_{dq} \in \{0, 1\} \quad \forall d \in D, \forall q \in Q_d \tag{29}$$

In terms of big O, this version of R_{dq}-routing model with a hypothesis relaxation has $O(|D| \cdot |Q| \cdot |R_{dq\max}| \cdot |H|)$ variables and $O(|D|^2 \cdot |Q|^2 \cdot |E| \cdot |R_{dqe\max}|^2 \cdot |H|)$ constraints. The changes made with the relaxation of the hypothesis in this section did not alter the complexity of the R_{dq}-routing model of Section III-F as the complexity of the extra variables and constraints proposed in this section was dominated by the complexity of the variables and constraints of the R_{dq}-routing model before the relaxation of the hypothesis.

IV. COMPUTATIONAL RESULTS

In this section, we comment on the results of the computational experiments we carried out on a wide range of instances. We run the program in C++ on Visual Studio 2019 (v142) editor using the CPLEX solver (version 20.1.0) on an Intel(R) Xeon(R) W-2123 CPU 3.60GHz 3.60 GHz with a 32GB RAM.

A. TEST INSTANCES

We tested the model with instances covering two types of data: the source and destination points' routes' matrix and the loading and stock models' plans. In the latter, all demands, as required by the industrial interlocutor, are planned for a planning time horizon of 15 days divided into units of 1-hour $Hmax = 360$ hours.

The data relating to the routes' matrix include three sets of routes whose numbers of routes amount to 55, 105, and 1590 routes, respectively.

The data relating to the loading and stock models constitute plans generated by the loading and stock suboptimizers and must be met by the routing model. The data used are extracted from loading/stocking suboptimizers.

The main data include:

- The available periods within the 15-day planning horizon. All vessels' demands have three available periods: from hour 1 to hour 25, from hour 55 to hour 140, and from hour 200 to hour 360, respectively. The demands of trucks and stocks have all one available period: the 15-day interval.
- The handling time for vessels' loading depends on the length of the vessel to be served. It follows the uniform distribution U[12,24] hours and U[19,40] hours for small and large vessels, respectively. The handling time of trucks loading and stock varies from 10 to 20 hours and 10 to 40, respectively, as collected from the studied case.
- The total number of qualities of a demand is the sum of the number of successive qualities served by all handling equipment at a time. It is also the maximal number of recipient spaces to house the product. It amounts to 5 in large vessels and stocks' demands, 2 in small vessels, and 1 in trucks' demands.
- Six sets of demands' instances. Three have a number of demands totaling 8, 16, and 24, respectively, for vessels' demands. The remaining three sets have a number of demands amounting to 14, 26, and 38, respectively, for the combination of vessels, trucks, and stocks' demands. 16 demands, including a mix of vessels, trucks, and stocks' demands over the 15-day planning period, presents the real max number of demands.
- The transfer time is at most 1 hour. Hence, in later sections, we report the computational experiments of the R-routing and R_{dq}-routing models. For the paper's sake, the computational results of the relaxed R_{dq}-routing model are not reported.

For each set of demands, a set of seven types of instances are considered for testing the different combinations of parallel and successive qualities handling for small and large vessels:

1) $\gamma_d = 1, nQ_d^1 = 1$ (small and big vessels)
2) $\gamma_d \leq 2, nQ_d^1 = 1, (nQ_d^2 = 1$ iff $\gamma_d = 2)$ (small and big vessels)
3) $\gamma_d = 2, nQ_d^1 = 1, nQ_d^2 = 1$ (small and big vessels)
4) $\gamma_d = 1, nQ_d^1 > 1$ (small vessels)
5) $\gamma_d \leq 2, \begin{cases} nQ_d^1 = 2, & \gamma_d = 1 \\ nQ_d^1 = 2, nQ_d^2 = 1, & \gamma_d = 2 \end{cases}$ (small vessels)
6) $\gamma_d = 1, nQ_d^1 > 1$ (large vessels)
7) $\gamma_d = 2, 2 \leq nQ_d^1 \leq 3, 1 \leq nQ_d^2 \leq 2$ (large vessels)

For demands' instances of vessels, stocks, and trucks, all seven instances remain the same for vessels. All trucks' demands added are featured with $\gamma_d = 1, nQ_d^1 = 1$ in all seven instances. The stocks' demands added have $\gamma_d = 1, nQ_d^1 = 1$ only in instances (1), (2), (3) and have $\gamma_d = 1, 2 \leq nQ_d^1 \leq 5$ in instances (4), (5), (6) and (7).

It is worth noting that (1) represents the extremely easy type of instances while (7) represents the extremely difficult type of instances that are unlikely to occur in reality.

All tested instances along with their results are reported in an online data set whose link is given in Section VI.

B. ILLUSTRATIVE EXAMPLE

In this section, we run the routing model on the seventh instance (I7-14-55) of 14 demands (including vessels, stocks, and trucks) over the 15-day planning period. The result is supposed to present a routing program of these 14 demands while assigning to them routes from the set of 55 routes. **Table 2** presents the main data used in the instance, including the earliest transport start date and the latest end date of each demand. The model is set to compute the adequate transport start and end dates according to the chosen routes while satisfying the model's constraints. **Table 3** presents fixed parameters for vessels, stocks, and trucks' demands given in the instance under illustration. For instance, the number of handling equipment for each vessel demand equals two. In contrast, it is set to one for stocks and trucks' demands. The whole [1, 360] time interval represents the available period for stocks and trucks. At the same time, there are three available periods for vessels, namely [1 - 25], [55 -140], and [200 - 360]. The pre-handling duration equals 10 and 1, respectively, for vessels' demands and stocks' and trucks' demands. **Fig. 7** shows the optimal transportation plan of instance I7-14-55 after running the routing model.

The legend in **Fig.7** describes unavailability periods for vessels in red-hatched zones. The demands are supposed to be presented in such a way that no demand starts before the time value (start hour flag) obtained out of the sum of the start of its available period plus its pre-handling duration minus the ceil of the expected transfer time, nor finishes by the time value (end hour flag) obtained out of the difference between the end of its available period and the post-handling duration. The vessels' start and end hour flags are illustrated in **Fig.7** in yellow and purple, respectively. At the same time, the stocks and trucks' values are not presented as they equal the ends of the planning horizon (1 and 360, respectively). Each blue and pink (if any) rectangle refers to the transport time slot of a quality served by the first and the second (if any) handling equipment, respectively. The subsequent rectangles denote the time slots of successive qualities. For each demand, the start of the first time slot marks its earliest transport start date. In comparison, the end of the last time slot marks its earliest transport end date. Besides, the red dots mark the latest end date, which, if exceeded, the demand gets tardiness.

However, the risk for tardiness is minimal if the total margin (difference between the latest transport end date and earliest transport start date) is significant, as in the case of the demand of truck 3 in **Table 2**. The model can choose to keep the start date of transport of a demand as reported in **Table 2** or to postpone it to a later date. If this postponement does cause the rectangles (time slots) to exceed the red mark (latest end time), then only waiting time is generated. The time slots are cross-hatched in this case, as illustrated in the legend. However, if they exceed the latest end date, tardiness and waiting time are generated. In this case, the rectangles are dashed. In the legend, we can also find the notation $\mathrm{x}nQ_d^{1,2}$, which denotes how many successive qualities are transported from a source to a destination. This destination is none other than the handling equipment. If two handling equipment are assigned to a demand, two routes are assigned to transport $\mathrm{x}nQ_d^1$ qualities, denoted in blue, and $\mathrm{x}nQ_d^2$ qualities, denoted in pink. Finally, the precedence table is also presented in the legend. It informs about the precedence between two demands, distinguishes between the predecessor and the successor demand, and the time interval between them $\Delta_{dd'}$.

As already mentioned, any forward displacing in the time axis means a waiting time was generated, thereby leading to some waiting costs being incurred. This is the case for four vessels 1, 2, 5, and 6. The waiting time in vessel 6, in particular, led the transportation service to exceed the latest end time, which brought about a tardiness duration and consequently entailed a tardiness penalty in addition to its waiting cost. The sum of these costs made up an objective value of 667.5. The details of this calculation are given in **Table 4**.

The time slots of vessel 2 got shifted because of vessel 3. Both vessels have overlapping routes during overlapping time slots. The decision to delay the transportation service for vessel 2 rather than that for vessel 3 is relevant for two reasons. First, the opposite would end up having routes and time overlap of vessel 3 with vessel 1, whose time slots are near three critical time thresholds: the latest end date, the end hour flag (in purple), and the end of the planning horizon, which would prevent either vessel 1 or 3 from being served during the same planning horizon. Second, the tardiness penalty is 500 units for vessel 3 against 50 units for vessel 2. As a result, Vessel 1 got a waiting time mainly because of the 1-hour of vessel 2 precedence. Almost the same scenario happened with vessels 5 and 6. Both got delayed in the interest of vessel 7 because of routes overlaps between vessels 5 and

TABLE 2. Main data used in instance I7-14-55 to transport 14 demands on 55 routes.

Demand	Earliest start date	Latest end date
Vessel 1	302	356
Vessel 2	278	356
Vessel 3	251	356
Vessel 4	99	130
Vessel 5	96	136
Vessel 6	64	94
Vessel 7	64	136
Vessel 8	299	356
Stock 1	5	64
Stock 2	137	180
Stock 3	170	208
Truck 1	2	29
Truck 2	30	52
Truck 3	53	359

TABLE 3. Fixed parameters for vessels and for stocks and trucks'demands used in I7-14-55 to transport 14 demands on 55 routes.

	Demands	
	Vessels	Stocks and Trucks
Pre-handling duration	10	1
Post-handling duration	4	0
Number of handling equipment	2	1
Available periods	[1 - 25] [55 - 140] [200 - 360]	[1 - 360]

TABLE 4. Objective value computation of I7-14-55.

Demands	Vessel 1	Vessel 2	Vessel 5	Vessel 6
Waiting time (min)	7	6	5	11
Waiting cost (cost unit)	12	12	12	12
Tardiness (min)	0	0	0	3,55
Tardiness penalty (cost unit)	100	50	90	90
Total objective value	667,5			

7 and between vessels 6 and 7 during overlapping time slots (**Table 2**). These overlap problems were overcome by the decision made (**Fig. 7**). If vessel 7 were delayed rather than vessel 6, the time slots of vessel 7 would be hugely overlapping with the time slots of vessel 5, which are nonetheless near the next unavailable period. Thus, either vessel 7 or vessel 5 would exceed the latest end time and incur tardiness penalties.

For the sake of the paper, the solutions on the 105 and 1590 routes' instances could not be illustrated. However, as reported in **Table 6** and **Table7**, their objective values are smaller than those of the illustrative example of 55 routes. This reduction is due to the decrease of possible routes' overlap as the number of possible routes to choose from increases.

The *R*-routing model could not generate a solution for the illustrative example instance, even for 55 routes, because of insufficient memory (**Table 8**). Thanks to the R_{dq}-routing model, we can propose routes that respect routing constraints

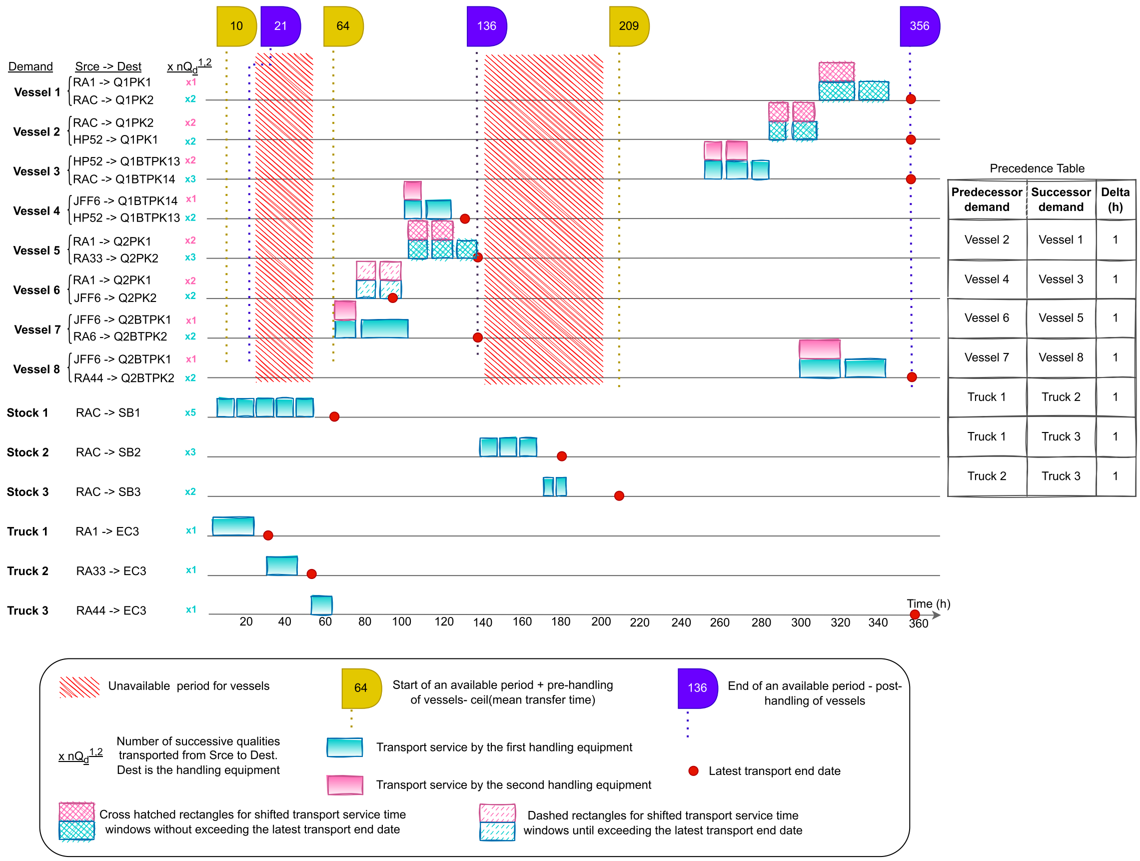


FIGURE 7. Transport plan of 14 demands after running the R_{dq}-routing model on 55 routes.

for different demands considering the conditions set by the loading and stock constraints. Finally, as the number of routes increases, the model is unlikely to find routes overlapping, or it may have access to routes with shorter transfer times.

C. EVALUATION OF THE MODELS

In this section, we assess the computational performance of the proposed models by verifying three objectives of interest. First, to check whether the models can be solved in a reasonable time using commercial software. Second, to observe how some parameters (the number of demands, the number of routes, and the combination of the number of handling equipment used at a time with the number of successive qualities) affect the computation time. Finally, to explore ways to accelerate the computation time.

For each model, we considered different groups of instances regarding the number of routes, the number of demands, and the nature of demands (vessels only or a combination of vessels, stocks, and trucks' demands). Each group of demands has a set of 7 instances to test the parallel and successive configurations for small and large vessels, as described in the previous section. As a result, we had 126 scenarios to test for each model (7 combinations of successive and parallel qualities × 3 sets of demands' number × 2 kinds of demands' nature × 3 routes' instances).

1) EVALUATION OF THE R_{dq}-ROUTING MODEL

The results obtained for the R_{dq}-routing model with a time limit of 6 hours are reported in **Table 5**, **Table 6**, and **Table7** for the 55-route, 105-route, and 1590-route instances, respectively. They all have two main columns referring to vessels' demands only and vessels, stocks, and trucks' demands, respectively. Each of these two main columns has three main sets of rows referring to the number of demands treated. In each row of each rows' set, the instance is presented as Ii-d-routes' number, such as "i" is the i-th combination of successive and parallel configurations among the seven existing combinations. Considering the three rows' sets of demands, i ranges from 1 to 21. "d" is the number of demands in question, (8, 16, 24) demands for vessels' demands only, and (14, 26, 38) demands for vessels, stocks and trucks' demands. Finally, the routes' number refers to the routes' set instance in question. The results given for each instance are shown in the following four columns representing the objective value, the execution time in minutes, and the gap in percentages provided by CPLEX. The average gap is also reported in the fifth column for a whole set of demands. The average gap equals the sum of the individual gap values divided by the number of solved instances, with the unsolved instances not included in the calculation. The results' fields with the (-) character indicate insufficient memory. The results unfold many observations, such as:

The increase in demands' number leads to an increase in the computation time for all scenarios. Besides, raising the number of routes for a fixed number of demands also increases computation time for all scenarios. It can even be seen that the increase of the computation time is shaper when we increase both the number of demands along with the number of routes, especially with instances where the parallel configuration is used in all vessels' demands (see for instances I3, I7, I10, I14, I17, and I21). Therefore, the computation time is directly impacted by high congestion in terms of demands to be satisfied and the complexity of the routes' matrix of the port. Notably, if any of these routes operate simultaneously, i.e., in a parallel configuration of a vessel, that causes two routes to be operating simultaneously. If the parallel configuration is applied to all vessels, there will be a need for vessels' number to the power of two as a total of routes to handle, in addition to routes transporting for stocks and trucks, if any.

Another observation is that the generated routing solutions are optimal using the real routes' matrix (1590) and up to the real max number of demands (16). For instance, for 8 demands, all given solutions are optimal with a maximal computation time of about 2 hours. For 14 demands, all instances are solved to optimality with a maximal computation time of 4 hours. For 16 demands, 6 out of 7 solutions are optimal, while the 7th is not generated because of memory insufficiency. This type of instance is very complex, where all the demands have parallel and successive configurations at a time. As one can notice, in **Table 5** and **Table 6**, all instances with a number of demands smaller than or equal to 16 are found with null gaps, unless the instance I14-16 has a gap of 45.29% and 76.72%, respectively. This shows the difficulty of this particular instance compared to other instances. Thus, when expanding the matrix of routes, it is evident that the complexity increases accordingly. As mentioned earlier, this type of instance has a feeble possibility of occurrence as it is improbable to have for each demand of the whole demands, parallel and successive configurations at a time. Apart from this particular instance, all other instances with up to the maximum expected number of demands were solved optimally in a reasonable time. Regarding higher demands (24, 26, and 38), gaps start to appear in 55 routes' set (with a maximum gap of 83.04%) and increase in 105 routes' set (with a maximum gap of 99.33%) to end up in the real set of routes with non-generated solutions. The results show that most of the solutions obtained are optimal (71%), knowing that higher demands were tested to reach the limit of the model.

We also observe that the objective values increase, which means the port incurs more costs as the number of demands increases and as the number of routes decreases. In fact, the more demands, the more routes overlap can occur between demands. However, the more possible routes the model can choose from, (*i*) the less overlap can appear so that most demands can occur simultaneously without incurring consequent costs, and/or (*ii*) the less transfer time can be obtained for a better solution. The most significant objective values in the three routes' sets are in instances where we assign the parallel configuration in all vessels' demands (Instances I3, I7, I10, I14, I17, I21), especially if all vessels are large and undergo successive qualities configuration as well (I7, I14, I21) independently of the number of demands.

On the other hand, one can observe that the increase in objective values (waiting and tardiness costs) with a growing number of demands and descending number of routes has another vital insight. An insight that we find promising regarding integration with the other operations' optimizers. In fact, in **Table 7**, where solutions using the real matrix of routes are reported (1590), the first set of rows show objective functions with null costs, unless in two instances where only small waiting costs are detected while tardiness penalties were not incurred. This means that the industrial manager can use the routing suboptimizer for up to 16 demands (a mix or vessels only) while conserving the solutions given by the loading/stocking suboptimizers. It is worth noting that the waiting costs (contrary to tardiness penalties) are artificial expenses that do not exist in the real case study but were added to avoid any unnecessary changes in the start demand dates specified by the loading/stocking plan. For the second set of rows, null costs are still observed in all instances the solver can handle. When the demands get higher than expected, tardiness costs appear in the few instances solved. This means that the plan given by the other suboptimizers is no longer preserved completely.

In summary, the proposed R_{dq}-routing model can be solved optimally in a reasonable time using an average set of demands while conserving the overall plan suggested by the loading/stocking suboptimizers. The thing that will set the base for integrating our model into the whole system without extra costs.

2) EVALUATION OF THE R-ROUTING MODEL

We could not obtain solutions for all instances with the Rrouting model. Only six among 126 instances were solved to optimality (5%) (see **Table 8**). All remaining instances constituted a memory problem for the model. As **Table 8** shows, not only could the R_{dq}-routing model solve way more complex instances, but these six instances are also solved optimally by the R_{dq}-routing model in an execution time that does not attain a tenth of a minute (0.1). In contrast, the R-routing model takes an execution time that ranges from 37 min to 327 min. Hence, the importance of the R_{dq}-routing model.

V. MANAGERIAL INSIGHTS AND RESEARCH OPPORTUNITIES TOWARDS REAL-TIME PLANNING/CONTROL INTEGRATION

What is interesting about studying the problems in an integrated fashion, whether by formulating a simultaneous optimization or by adopting the coupling strategy between

TABLE 5. Solving the R_{dq}-routing model on 55 routes over a 15-day period with a time limit of 6 hours.

Vessels' demands only					Vessels, stocks and trucks' demands				
Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Avg. gap (%)	Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Avg. gap (%)
I1-8-55	0	0.01	0.00		I1-14-55	0	0.07	0.00	
I2-8-55	0	0.03	0.00		I2-14-55	0	0.09	0.00	
I3-8-55	252	0.06	0.00		I3-14-55	252	0.19	0.00	
I4-8-55	0	0.02	0.00	0.00	I4-14-55	0	0.13	0.00	0.00
I5-8-55	0	0.03	0.00		I5-14-55	0	0.19	0.00	
I6-8-55	0	0.05	0.00		I6-14-55	0	0.25	0.00	
I7-8-55	667.5	0.27	0.00		I7-14-55	667.5	1.51	0.00	
I8-16-55	0	0.06	0.00		I8-26-55	0	0.40	0.00	
I9-16-55	14705	0.49	0.00		I9-26-55	14705	1.12	0.00	
I10-16-55	60293.8	66.25	0.00		I10-26-55	60590.6	167.39	0.00	
I11-16-55	0	0.09	0.00	6.47	I11-26-55	0	0.51	0.00	8.87
I12-16-55	8573.1	0.67	0.00		I12-26-55	8573.1	2.69	0.00	
I13-16-55	393	0.96	0.00		I13-26-55	393	2.03	0.00	
I14-16-55	70184.6	360.11	45.29		I14-26-55	74080.5	360.17	62.06	
I15-24-55	9269.5	0.32	0.00		I15-38-55	9726.5	1.71	0.00	
I16-24-55	57699	7.93	0.00		I16-38-55	60722.3	38.11	0.00	
I17-24-55	112348	360.08	13.08		I17-38-55	119352	360.11	27.04	
I18-24-55	2542	1.01	0.00	14.40	I18-38-55	3735.2	39.74	0.00	21.54
I19-24-55	56288.4	360.06	4.71		I19-38-55	62230.8	360.13	19.91	
I20-24-55	69726.7	79.35	0.00		I20-38-55	74311.5	360.19	24.30	
I21-24-55	217234	360.43	83.04		I21-38-55	194114	360.22	79.54	

TABLE 6. Solving the R_{dq}-routing model on 105 routes over a 15-day period with a time limit of 6 hours.

Vessels' demands only					Vessels, stocks and trucks' demands				
Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Avg. gap (%)	Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Avg. gap (%)
I1-8-105	0	0.03	0.00		I1-14-105	0	0.19	0.00	
I2-8-105	0	0.11	0.00		I2-14-105	0	0.36	0.00	
I3-8-105	0	0.33	0.00		I3-14-105	0	0.81	0.00	
I4-8-105	0	0.05	0.00	0.00	I4-14-105	0	0.44	0.00	0.00
I5-8-105	0	0.15	0.00		I5-14-105	0	0.64	0.00	
I6-8-105	0	0.16	0.00		I6-14-105	0	0.74	0.00	
I7-8-105	72	0.46	0.00		I7-14-105	72	3.16	0.00	
I8-16-105	0	0.39	0.00		I8-26-105	0	1.16	0.00	
I9-16-105	12366	1.77	0.00		I9-26-105	12366	3.56	0.00	
I10-16-105	36823.2	360.12	6.36		I10-26-105	36823.2	360.15	10.60	
I11-16-105	0	0.48	0.00	11.87	I11-26-105	0	2.01	0.00	12.28
I12-16-105	252	1.41	0.00		I12-26-105	1224.8	31.23	0.00	
I13-16-105	0	1.85	0.00		I13-26-105	0	4.72	0.00	
I14-16-105	31083.6	360.27	76.72		I14-26-105	47123.3	360.39	75.39	
I15-24-105	738	0.62	0.00		I15-38-105	1243	3.24	0.00	
I16-24-105	34751	40.70	0.00		I16-38-105	37770	129.85	0.00	
I17-24-105	72982.5	360.23	47.03		I17-38-105	97156.7	360.30	83.23	
I18-24-105	734	1.50	0.00	23.37	I18-38-105	4394.4	360.30	82.58	51.90
I19-24-105	36945	24.95	0.00		I19-38-105	38258.2	360.28	1.01	
I20-24-105	53510.6	360.30	17.21		I20-38-105	186797	360.48	96.90	
I21-24-105	382893	360.57	99.33		I21-38-105	441218	360.74	99.60	

separate problems' optimizers, namely the routing optimizer of this paper with constraints of interaction with other problems, is that even managerial insights are derived considering the concordance and consistency between the different bulk port problems. In fact, the planning operations' problems need to be aligned around the port supply-chain objective, ensuring a positive impact on the bottom line not only on an operational or tactical level but also on a strategic level. Because if efforts are being carried out on the strategic level without looking for optimal supply-chain design decisions, the operational or tactical optimization would be difficult to reach and unable to adhere to the strategic planning without hindering the group's ultimate mission; maximizing loaded volume and minimizing time penalties. And this scenario would compel the group to reconsider strategic planning for correction and adaption.

With that being said, one can note, from the routing models' results, that the transportation routes' facility is better off with many routes as this improves the objective value even if the computation time may increase. The latter is acceptable for tactical decisions made for more than a week (15 days in our case). The increase in routes' number is beneficial for

TABLE 7. Solving the R_{dq}-routing model on 1590 routes over a 15-day period with a time limit of 6 hours.

Vessels' demands only					Vessels, stocks and trucks' demands				
Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Avg. gap (%)	Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Avg. gap (%)
I1-8-1590	0	6.43	0.00		I1-14-1590	0	8.38	0.00	
I2-8-1590	0	25.80	0.00		I2-14-1590	0	23.64	0.00	
I3-8-1590	0	55.77	0.00		I3-14-1590	0	80.59	0.00	
I4-8-1590	0	1.83	0.00	0.00	I4-14-1590	0	4.77	0.00	0.00
I5-8-1590	0	13.12	0.00		I5-14-1590	0	19.94	0.00	
I6-8-1590	0	8.50	0.00		I6-14-1590	0	19.24	0.00	
I7-8-1590	12	125.04	0.00		I7-14-1590	12	239.54	0.00	
I8-16-1590	0	25.41	0.00		I8-26-1590	0	30.94	0.00	
I9-16-1590	0	74.17	0.00		I9-26-1590	0	107.31	0.00	
I10-16-1590	0	177.53	0.00		I10-26-1590	-	-	-	
I11-16-1590	0	27.60	0.00	0.00	I11-26-1590	0	43.49	0.00	0.00
I12-16-1590	0	93.36	0.00		I12-26-1590	0	242.83	0.00	
I13-16-1590	0	371.85	0.00		I13-26-1590	-	-	-	
I14-16-1590	-	-	-		I14-26-1590	-	-	-	
I15-24-1590	534	80.49	0.00		I15-38-1590	1039	183.10	0.00	
I16-24-1590	-	-	-		I16-38-1590	-	-	-	
I17-24-1590	-	-	-		I17-38-1590	-	-	-	
I18-24-1590	105350	381.52	99.63	49.82	I18-38-1590	-	-	-	0.00
I19-24-1590	-	-	-		I19-38-1590	-	-	-	
I20-24-1590	-	-	-		I20-38-1590	-	-	-	
I21-24-1590	-	-	-		I21-38-1590	-	-	-	

(-) : Out of memory.

TABLE 8. Obtained results with the *R*-routing model in comparison to their equivalent obtained with the *Rdq*-routing model.

	R-routing model			R_{dq}-routing model		
Ins.	Obj. value	Time (min)	CPLEX Gap (%)	Obj. value	Time (min)	CPLEX Gap (%)
I1-8-55	0	36.66	0.00	0	0.01	0.00
I2-8-55	0	122.22	0.00	0	0.03	0.00
I4-8-55	0	102.30	0.00	0	0.02	0.00
I8-16-55	0	369.93	0.00	0	0.06	0.00
I1-14-55	0	327.30	0.00	0	0.07	0.00
I1-8-105	0	228.34	0.00	0	0.03	0.00

the routing and the other bulk port optimization problems as there will be more optional routes to operate in parallel without any problem, thus serving a more significant number of demands and transporting a greater volume of product. However, it is legitimate to devote an in-depth reflection to whether the increase of the number of routes should be done by adding routes between the existing number of sources and destinations (if we keep the reasonable maximal number of demands as 16) or by adding the number of sources and/or destinations, thus creating additional routes. It is reasonable to think that adding sources or/and destinations will be helpful whether we keep the same number of demands or increase it.

Nonetheless, one must pay attention to how the number of sources or/and destinations should be increased. This is because adding a source may refer to adding a whole storage shed with all its equipment (stacker and scraper) or to adding only a scraper to the existing scraper and stacker of the existing storage shed. Option one may incur costs, but option two may bring infeasibility, obstruction, or (at best) low performance in port logistics operations. There are two reasons why adding a scraper inside an existing storage shed may hinder port operations; (*i*) Within a storage shed, the stacker and the scraper cannot operate close to each other. Therefore, adding a second scraper may further constrain the scraping or stocking operations. At best, the scraping strategy would be vertical, which entails a low performance of scraping flow rate. But in many scenarios, either stocking or one scraping operation among the two scraping operations would be blocked. (*ii*) A scraper is better off not making many direction changes as this would play against the maximization of throughput. Indeed, by changing the direction of the scraper each time, there is a risk of losing the pace of the cadence. As to adding a destination, it depends on stocking or loading demands. For stocking demands, adding a destination may refer to adding a whole storage shed with all its equipment (stacker and scraper) or to adding only a stacker to the existing stacker of the existing storage shed. As a result, the same points reported in adding a source also apply here. For loading demands, adding a destination may refer to adding a whole quay with an additional number of loaders or to adding only the equipment (gantry cranes) for vessels' loading. Fortunately, this time building a whole new quay is not necessary because increasing the number of gantry cranes in a quay does not constrain the functioning of any port equipment in any way, neither in terms of its availability nor in terms of its performance, apart from a minimal security distance needed between two consecutive gantry cranes. Thus far, adding routes can be carried out depending on the number of demands the group aims to satisfy in a 15-day horizon; adding routes between existing sources and destinations if the number of demands does not exceed 16. The objective would be to ensure a maximal performance in terms of minimal tardiness and waiting times. But if the number of demands must increase, adding routes between additional sources and destinations would be salutary by taking into account the measures and risks of doing so (already reported in this section). We could draw insights on problems and operations other than routing because of the takeaway ideas we could get from the integration architecture we proposed in our former work [6].

In discussing the aimed integration approach, we proposed that OCP managers start optimization of the loading problem (See **Fig. 2**). It is the first problem launched because the maximal loaded volume and minimal time penalties are the strategic objectives of OCP. Then, the other operations follow. So that the loading operation could be carried out on time, products' stockpiles must be stocked, cooled, and equalized, which is a necessary condition for the scraping to begin, thus transportation as well. And so that the product can be stocked on time, it must be produced on time. Hence, OCP would launch production following this strategy to avoid over-production and produce only what is needed and what was ordered. In this paper, we developed the routing problem by considering this vision of planning problems integration.

However, the debate is not settled here because the resilience needed to use port operations efficiently requires the integration of planning problems with the control system so that the optimization result serves as setpoints for the real industrial setting to execute the optimization plan on physical equipment. Another reason behind the integration with the control system is that disturbances, feedback cycles, and dynamics such as equipment breakdowns and performance degradation are considered because an industrial setting is not an ideal environment [6]. However, it is prone to perturbations that must be corrected and trigger re-planning if necessary. Overall, integrating the optimization tool within the control system should be a must and a high-priority requirement. The same need was expressed in [48], in which it is reported that continuous feedback and re-planning in case of production deviations is compulsory for a planning tool to be set up in autonomous industrial plants. The latter is becoming a necessity that must be achieved rather than a luxury. The architecture presented in **Fig. 8** is none other than the architecture of integration of the operations' optimization tool with the control system using some extra enabler functionalities that we propose as future research opportunities:

- The prediction tool predicts the equipment state (flow rate and availability) to help the planning tool decide on a better-quality solution. These data feed a common database that, in turn, feeds an internal unit to the planning tool, which is a preprocessing unit. The latter prepares data for each loading, stocking, routing, and production subsystem, providing them with information performance and parameters of available equipment and routes.
- The digital twin helps the user choose the most suitable plan from the plans offered by the planning tool, as the planning generates a set of integrated optimal or near-optimal plans because of the multi-objective aspect of the loading suboptimizer.
- The recovery tool that deals with disturbances and coordinates with the control system and planning tool for complete command of the industrial setting. In fact, in case of disturbances, it has to decide whether a re-planning request should be sent to the planning tool for replanning or it only adjusts the plan without soliciting the planning tool but sends the adjusted plan to the control system for execution. Since re-planning each time, a disturbance occurs can be time-consuming. As a result, the control system would never execute the plan as the latter keeps getting updated. But again, not re-planning would lead the port operations in significant drifts.
- The unit of re-planning test compares the re-planned plan of the planning subsystems with the adjusted plan returned by the recovery and decides upon the best solution. This is an internal unit of the optimization tool (not illustrated in **Fig. 8**).
- Tactical-operational interface to process or interpret data of optimization tool (tactical level) for the control system (operational level). This is because, in the tactical level plan, data are aggregated. While the control system and the digital twin need more detailed data to execute the plan at the operational level.

The architecture of **Fig. 8** incorporates a high degree of intelligence and autonomy required to respond to environmental changes, ensuring multidisciplinary integration, which is undervalued in optimization tools, but also in traditional mechatronic systems. The investment in such architecture is relevant because the integration of planning and control systems is not feasible if we rely on the traditional mechatronic systems as they are independent and consider humans outside the loop, nor if we rely on optimization tools with state-of-the-art algorithms that are developed in isolation of the real industrial world and its disturbances and dynamics. We could go as far as to say that before the advent of industry 4.0 concepts, the integration problem was more a theoretical issue than a practice. The integration could not become a reality because of the challenges that emerge from the heterogeneity, concurrency, and timing responsiveness aspects of the whole integrated system. In the two levels of integration; (*i*) the integration of optimization problems and (*ii*) the integration of the integrated optimization tool

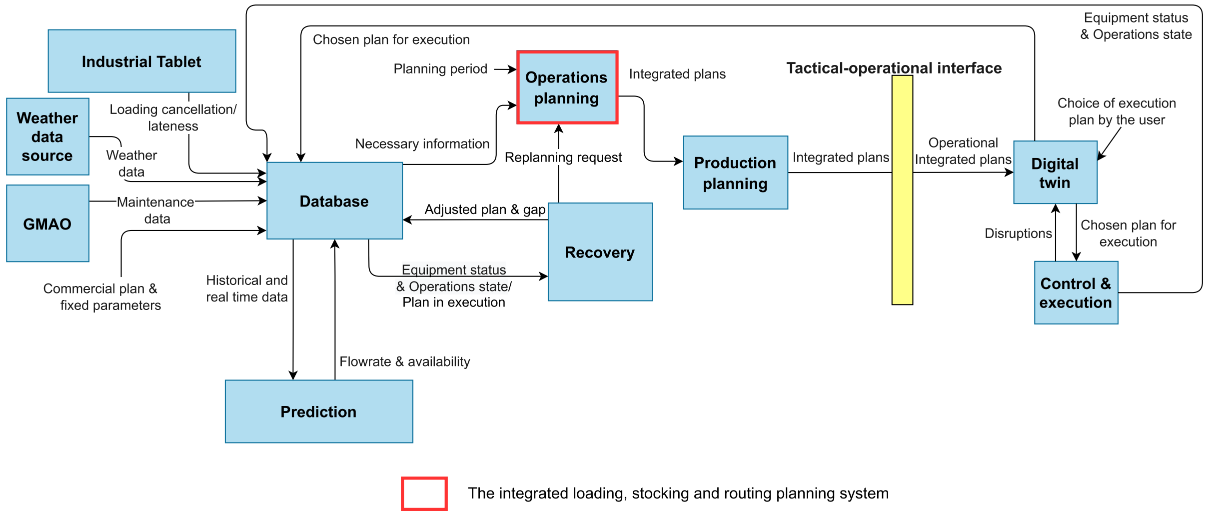


FIGURE 8. **Integrated planning (optimization) and control system from [6].**

with the control system, we dealt with complexity that had to be mastered and controlled with the help of the Systems Engineering approach [49], [50]. The latter is a scientific approach from Systems theory [51] to master complexity during the design phase and an ultimate approach for worldwide critical industries that enabled the integration in both levels in a smart fashion. To fulfill the Systems Engineering phases successfully, we had a collaboration with different stakeholders of OCP Jorf Lasfar bulk port in order to elicit and analyze all requirements for planning/control integration and for planning problems' integration and in order to validate the functional architectures, we proposed as a result of working through the elicited requirements, in our former paper [6].

VI. CONCLUSION

In this study, we advanced the field of bulk port optimization by offering a unique approach to solve the complex belt-conveyor routing problem within bulk ports. The problem presented numerous complexities due to interdependencies within the routing perimeter and interactions with other port operations, primarily loading and stocking.

Our novel approach towards modeling the routing problem was focused on optimization, considering both the constraints of routing and interactions, as well as parameters of other bulk port problems. Adopting this view in modeling allowed (*i*) a real-world synchronization of the transportation operation with other bulk port operations, which mirrored actual operations, and (*ii*) a successful first step towards the objective of integrating the bulk port problems.

Contrary to the conventional method of simultaneous optimization in a single mathematical model, our aimed integration approach involves optimizing bulk port problems separately, maintaining a coupling strategy to avoid contradictions and quasi-infinite loops between the problems' optimizers, and ensuring concordance. This groundbreaking method surpasses the previously achieved integration of a maximum of two problems without including critical decisions, as highlighted in our literature review.

Based on the scenario of the aimed integration approach, We proposed two distinct routing models: The R-routing and R_{dq}-routing models. These models optimally select routes with shorter transport time from either the complete set R or the subset R_{dq}, respectively, minimizing waiting costs and tardiness penalties. Notably, the R_{dq}-routing model showed significantly improved results compared to the R-routing model. These satisfactory results promise for a successful integration as found out in the experiments. The R_{dq}-routing model was further enhanced to provide a more general application to the model.

Despite the promising results, the bulk port belt-conveyor routing problem remains a complex optimization problem that requires a more sophisticated, hybrid approach. In the face of challenging instances, we suggest approximation methods, refining the mathematical model, and improving the solution using constraint programming (CP). This CP-MIP hybrid approach can further benefit from the integration of AI methods, enhancing control of computational performance and memory usage.

DATA AVAILABILITY STATEMENT

Data supporting these findings' study, including instances and results, are reported in the following online Data set (https://data.mendeley.com/datasets/m69nbt37tz/1).

REFERENCES

[1] United Nations, Geneva, Switzerland. (2021). *Review of Maritime Transport 2021 | UNCTAD, United Nations Conference on Trade and Development*. [Online]. Available: https://unctad.org/webflyer/review-maritime-transport-2021

[2] J. Xin, R. R. Negenborn, and T. van Vianen, "A hybrid dynamical approach for allocating materials in a dry bulk terminal," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1326–1336, Jul. 2018, doi: 10.1109/TASE.2017.2784483.

[3] A. Lim, "The berth planning problem," *Oper. Res. Lett.*, vol. 22, nos. 2–3, pp. 105–110, Mar. 1998, doi: 10.1016/S0167-6377(98)00010-8.

[4] P. Chen, Z. Fu, A. Lim, and B. Rodrigues, "Port yard storage optimization," *IEEE Trans. Autom. Sci. Eng.*, vol. 1, no. 1, pp. 26–37, Jul. 2004, doi: 10.1109/TASE.2004.829412.

[5] B. Santosa, R. Damayanti, and B. Sarkar, "Solving multi-product inventory ship routing with a heterogeneous fleet model using a hybrid cross entropy-genetic algorithm: A case study in Indonesia," *Prod. Manuf. Res.*, vol. 4, no. 1, pp. 90–113, Jan. 2016, doi: 10.1080/21693277.2016.1204961.

[6] S. Mallah, A. Aloullal, O. Kamach, K. Kouiss, N. Najid, and L. Deshayes, "A novel integration approach for a complex supply chain optimization problem in an export bulk port," in *Proc. 7th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Jun. 2020, pp. 751–756, doi: 10.1109/CoDIT49905.2020.9263911.

[7] M. Hayashi, K. Higuma, Y. Miyamoto, S. Ihara, and S. Edo, "A solution method for an assignment and routing problem of vehicles for a container yard," *Trans. Soc. Instrum. Control Eng.*, vol. 40, no. 11, pp. 1140–1147, 2004, doi: 10.9746/SICETR1965.40.1140.

[8] K. H. Kim and K. Y. Kim, "Routing straddle carriers for the loading operation of containers using a beam search algorithm," *Comput. Ind. Eng.*, vol. 36, no. 1, pp. 109–136, Jan. 1999, doi: 10.1016/S0360-8352(99)00005-4.

[9] E. Nishimura, A. Imai, and S. Papadimitriou, "Yard trailer routing at a maritime container terminal," *Transp. Res. E, Logistics Transp. Rev.*, vol. 41, no. 1, pp. 53–76, Jan. 2005, doi: 10.1016/J.TRE.2003.12.002.

[10] M. Zhong, Y. Yang, S. Sun, Y. Zhou, O. Postolache, and Y.-E. Ge, "Priority-based speed control strategy for automated guided vehicle path planning in automated container terminals," *Trans. Inst. Meas. Control*, vol. 42, no. 16, pp. 3079–3090, Jul. 2020, doi: 10.1177/0142331220940110.

[11] T. W. Liao, "Integrated inbound vehicle routing and scheduling under a fixed outbound schedule at a multi-door cross-dock terminal," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13217–13229, Aug. 2022, doi: 10.1109/TITS.2021.3122396.

[12] W. Wisittipanich and P. Hengmeechai, "Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization," *Comput. Ind. Eng.*, vol. 113, pp. 793–802, Nov. 2017, doi: 10.1016/J.CIE.2017.01.004.

[13] P. Avella, M. Boccia, and A. Sforza, "Solving a fuel delivery problem by heuristic and exact approaches," *Eur. J. Oper. Res.*, vol. 152, no. 1, pp. 170–179, Jan. 2004, doi: 10.1016/S0377-2217(02)00676-8.

[14] C. Tilk and A. Goel, "Bidirectional labeling for solving vehicle routing and truck driver scheduling problems," *Eur. J. Oper. Res.*, vol. 283, no. 1, pp. 108–124, May 2020, doi: 10.1016/J.EJOR.2019.10.038.

[15] R. Liu and Z. Jiang, "A constraint relaxation-based algorithm for the load-dependent vehicle routing problem with time windows," *Flexible Services Manuf. J.*, vol. 31, no. 2, pp. 331–353, Jun. 2019, doi: 10.1007/S10696-018-9323-0.

[16] L. Wu, Z. He, Y. Chen, D. Wu, and J. Cui, "Brainstorming-based ant colony optimization for vehicle routing with soft time windows," *IEEE Access*, vol. 7, pp. 19643–19652, 2019, doi: 10.1109/ACCESS.2019.2894681.

[17] X. Zhu, R. Yan, Z. Huang, W. Wei, J. Yang, and S. Kudratova, "Logistic optimization for multi depots loading capacitated electric vehicle routing problem from low carbon perspective," *IEEE Access*, vol. 8, pp. 31934–31947, 2020, doi: 10.1109/ACCESS.2020.2971220.

[18] Y. Shen, M. Liu, J. Yang, Y. Shi, and M. Middendorf, "A hybrid swarm intelligence algorithm for vehicle routing problem with time windows," *IEEE Access*, vol. 8, pp. 93882–93893, 2020, doi: 10.1109/ACCESS.2020.2984660.

[19] J. A. Cano, P. Cortés, J. Muñuzuri, and A. Correa-Espinal, "Solving the picker routing problem in multi-block high-level storage systems using metaheuristics," *Flexible Services Manuf. J.*, vol. 35, pp. 376–415, Feb. 2022, doi: 10.1007/S10696-022-09445-y.

[20] M. Haouassi, Y. Kergosien, J. E. Mendoza, and L. M. Rousseau, "The integrated orderline batching, batch scheduling, and picker routing problem with multiple pickers: The benefits of splitting customer orders," *Flexible Services Manuf. J.*, vol. 34, pp. 614–645, Jul. 2021, doi: 10.1007/S10696-021-09425-8.

[21] W. Guo, M. Ji, and H. Zhu, "Multi-period coordinated optimization on berth allocation and yard assignment in container terminals based on truck route," *IEEE Access*, vol. 9, pp. 83124–83136, 2021, doi: 10.1109/ACCESS.2021.3086185.

[22] W. Dai, X. Sun, and S. Wandelt, "Finding feasible solutions for multi-commodity flow problems," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 2878–2883, doi: 10.1109/ChiCC.2016.7553801.

[23] W. Dai, J. Zhang, and X. Sun, "On solving multi-commodity flow problems: An experimental evaluation," *Chin. J. Aeronaut.*, vol. 30, no. 4, pp. 1481–1492, Aug. 2017, doi: 10.1016/J.CJA.2017.05.012.

[24] S. Singh, "Flow constrained minimum cost flow problem," *OPSEARCH*, vol. 49, no. 2, pp. 154–168, Jun. 2012, doi: 10.1007/S12597-012-0069-2.

[25] M. A. Arostegui, S. N. Kadipasaoglu, and B. M. Khumawala, "An empirical comparison of Tabu search, simulated annealing, and genetic algorithms for facilities location problems," *Int. J. Prod. Econ.*, vol. 103, no. 2, pp. 742–754, Oct. 2006, doi: 10.1016/J.IJPE.2005.08.010.

[26] G. C. Menezes, G. R. Mateus, and M. G. Ravetti, "A branch and price algorithm to solve the integrated production planning and scheduling in bulk ports," *Eur. J. Oper. Res.*, vol. 258, no. 3, pp. 926–937, May 2017, doi: 10.1016/j.ejor.2016.08.073.

[27] D. Komljenovic, J. Paraszczak, K. Fytas, and C. Drebenstedt, "Methodology to calculate the effective reclaiming capacity of rail-mounted boom-type bucket wheel reclaimer and stacker/reclaimer," *Bulk Solids Handling*, vol. 24, no. 3, pp. 158–171, 2004.

[28] M. Ago, T. Nishi, and M. Konishi, "Simultaneous optimization of storage allocation and routing problems for belt-conveyor transportation," *J. Adv. Mech. Des., Syst., Manuf.*, vol. 1, no. 2, pp. 250–261, 2007, doi: 10.1299/jamdsm.1.250.

[29] T. Robenek, N. Umang, M. Bierlaire, and S. Ropke, "A branch-and-price algorithm to solve the integrated berth allocation and yard assignment problem in bulk ports," *Eur. J. Oper. Res.*, vol. 235, no. 2, pp. 399–411, Jun. 2014, doi: 10.1016/J.EJOR.2013.08.015.

[30] S. Pratap, Y. Daultani, M. K. Tiwari, and B. Mahanty, "Rule based optimization for a bulk handling port operations," *J. Intell. Manuf.*, vol. 29, no. 2, pp. 287–311, Feb. 2018, doi: 10.1007/s10845-015-1108-7.

[31] O. Unsal and C. Oguz, "An exact algorithm for integrated planning of operations in dry bulk terminals," *Transp. Res. E, Logistics Transp. Rev.*, vol. 126, pp. 103–121, Jun. 2019, doi: 10.1016/J.TRE.2019.03.018.

[32] Ö. Ünsal, "Reclaimer scheduling in dry bulk terminals," *IEEE Access*, vol. 8, pp. 96294–96303, 2020, doi: 10.1109/ACCESS.2020.2997739.

[33] D. Sun, Y. Meng, L. Tang, J. Liu, B. Huang, and J. Yang, "Storage space allocation problem at inland bulk material stockyard," *Transp. Res. E, Logistics Transp. Rev.*, vol. 134, Feb. 2020, Art. no. 101856, doi: 10.1016/J.TRE.2020.101856.

[34] I. Krimi, R. Todosijević, R. Benmansour, M. Ratli, A. A. El Cadi, and A. Aloullal, "Modelling and solving the multi-quays berth allocation and crane assignment problem with availability constraints," *J. Global Optim.*, vol. 78, no. 2, pp. 349–373, Oct. 2020, doi: 10.1007/S10898-020-00884-1.

[35] Y. Xia, X. Liu, and G. Du, "Solving bi-level optimization problems in engineering design using Kriging models," *Eng. Optim.*, vol. 50, no. 5, pp. 856–876, May 2018, doi: 10.1080/0305215X.2017.1358711.

[36] E. Samaniego and P. Novoa-Hernández, "A hybrid approach for solving dynamic bi-level optimization problems," *Computación Sistemas*, vol. 22, no. 2, pp. 639–656, Jul. 2018, doi: 10.13053/CYS-22-2-2557.

[37] U. P. Wen and S. T. Hsu, "Linear bi-level programming problems—A review," *J. Oper. Res. Soc.*, vol. 42, pp. 125–133, Feb. 1991, doi: 10.1057/jors.1991.23.

[38] E. Kieffer, G. Danoy, P. Bouvry, and A. Nagih, "Bayesian optimization approach of general bi-level problems," in *Proc. Genet. Evol. Comput. Conf. Companion*, Jul. 2017, pp. 1622–1629, doi: 10.1145/3067695.3082537.

[39] N. Hamta, M. A. Shirazi, and S. F. Ghomi, "A bi-level programming model for supply chain network optimization with assembly line balancing and push–pull strategy," *Proc. Inst. Mech. Eng. B, J. Eng. Manuf.*, vol. 230, no. 6, pp. 1127–1143, Feb. 2015, doi: 10.1177/0954405414564406.

[40] M. Kolyaei, A. Azar, A. R. Gatari, and M. D. Nayeri, "A bi-level optimization model for supply chain with incremental discount structure," *J. Ind. Manage. Perspective*, vol. 12, no. 1, pp. 9–44, Mar. 2022, doi: 10.52547/JIMP.12.1.9.

[41] M. Abbassi, A. Chaabani, L. B. Said, and N. Absi, "Bi-level multi-objective combinatorial optimization using reference approximation of the lower level reaction," *Proc. Comput. Sci.*, vol. 176, pp. 2098–2107, Jan. 2020, doi: 10.1016/J.PROCS.2020.09.246.

[42] E. G. Talbi, "A taxonomy of metaheuristics for bi-level optimization," in *Metaheuristics for Bi-Level Optimization* (Studies in Computational Intelligence), vol. 482. Berlin, Germany: Springer, 2013, pp. 1–39.

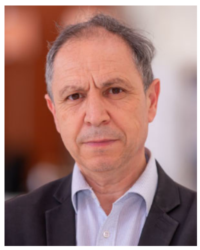
[43] Y. Marinakis and M. Marinaki, "A bilevel particle swarm optimization algorithm for supply chain management problems," in *Metaheuristics for Bi-Level Optimization* (Studies in Computational Intelligence). Berlin, Germany: Springer, vol. 482. 2013, pp. 69–93.
[44] F. Legillon, A. Liefooghe, and E. G. Talbi, "CoBRA: A coevolutionary metaheuristic for bi-level optimization," in *Metaheuristics for Bi-Level Optimization* (Studies in Computational Intelligence). Berlin, Germany: Springer, vol. 482. 2013, pp. 95–114.
[45] A. Koh, "A metaheuristic framework for bi-level programming problems with multi-disciplinary applications," in *Metaheuristics for Bi-Level Optimization* (Studies in Computational Intelligence), vol. 482. 2013, pp. 153–187.
[46] R. Oulhiq, K. Benjelloun, Y. Kali, and M. Saad, "A bi-level optimization approach for historical data-driven system identification," *J. Control, Autom. Electr. Syst.*, vol. 34, no. 1, pp. 73–84, Feb. 2023, doi: 10.1007/S40313-022-00951-W.
[47] F. Xhafa and A. Abraham, *Metaheuristics for Scheduling in Industrial and Manufacturing Applications* (Computational Intelligence), vol. 128. Berlin, Germany: Springer, 2008, p. 345. [Online]. Available: http://portal.acm.org/citation.cfm?id=1481665
[48] T. Gamer, M. Hoernicke, B. Kloepper, R. Bauer, and A. J. Isaksson, "The autonomous industrial plant -Future of process engineering, operations and maintenance," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 454–460, 2019, doi: 10.1016/J.IFACOL.2019.06.104.
[49] M. M. D. Hassan, "Engineering supply chains as systems," *Syst. Eng.*, vol. 9, no. 1, pp. 73–89, 2006, doi: 10.1002/SYS.20042.
[50] G. Locatelli, M. Mancini, and E. Romano, "Systems engineering to improve the governance in complex project environments," *Int. J. Project Manage.*, vol. 32, no. 8, pp. 1395–1410, Nov. 2014, doi: 10.1016/J.IJPROMAN.2013.10.007.
[51] K. M. Adams, P. T. Hester, J. M. Bradley, T. J. Meyers, and C. B. Keating, "Systems theory as the foundation for understanding systems," *Syst. Eng.*, vol. 17, no. 1, pp. 112–123, Mar. 2014, doi: 10.1002/SYS.21255.

SARA MALLAH is currently pursuing the joint Ph.D. degree with LTI, University Abdelmalek Essaadi, and GTI, Mohammed VI Polytechnic University (UM6P). She was a Research and Development Engineer with the Green Tech Institute (GTI). Her research interests include mathematical modeling, optimization techniques, and systems engineering to architect and manage complex systems. During the Ph.D. program, she presented three conference papers and one poster.

AFAF ALOULLAL is currently pursuing the Ph.D. degree with Universit Polytechnique Hauts-de-France. Previously, she was an Industrial Engineer with the Green Tech Institute (GTI), Mohammed VI Polytechnic University (UM6P). Her research interests include mathematical modeling and optimization techniques. During her work with GTI, she presented one conference paper and took part in other researcher's contributions (three conference papers and one article).

OULAID KAMACH received the Ph.D. degree in engineering science from INSA Lyon, in 2004, and the D.E.A. degree in applied mathematics from Besancon Franche Comte University. He is currently a Full Professor in industrial engineering and logistics with Abdelmalek Essaadi University. He is also the Head of Master Study on Management and Monitoring of Industrial System and an affiliate Professor with the Green Tech Institute (GTI), Mohammed VI Polytechnic University (UM6P).

MALEK MASMOUDI received the Doctorate of Sciences (Habilitation) degree from Jean-Monnet University, in 2017. He is currently pursuing the Ph.D. degree in industrial engineering. He was the Head of the Industrial Engineering Master with Jean-Monnet University. He is also an engineer. He is also an Associate Professor with the College of Engineering, University of Sharjah. He has published two springer books in data science and more than 27 international journal articles in operations research and industrial engineering. His research interests include the applications of data science and optimization techniques to various engineering and management fields.



KHALID KOUISS is currently a Professor in industrial computer sciences and industrial engineering with Mohammed VI Polytechnic University (UM6P) and the SIGMA Engineering School, Clermont-Ferrand. He is also an expert in industrial automation and systems engineering. His research interests include system modeling, systems engineering, intelligent manufacturing systems (IMS), manufacturing execution systems (MES), automation for complex manufacturing systems, industrial robotics, and more generally the deployment of industry 4.0 concepts.

AHMED CHEBAK received the Engineering Diploma degree in electromechanics from the National School of Mineral Industry in Rabat and the M.Sc. and Ph.D. degrees from Laval University, Canada. He is currently a Professor in electrical engineering with Mohammed VI Polytechnic University (UM6P) and the Pedagogical Director of the Green Tech Institute (GTI). He was a Full Professor with the University of Quebec at Rimouski, Canada, and a research associate and lecturer at Laval University. He was also a member of the Educational Committee of the Institute of Electrical Energy Engineering (IGEE) in Montreal. His research interests include the application of optimization in smart grids and industrial plants.