Robust ship pipe routing: navigating the energy transition

Berend Markhorst*[1,2], Joost Berkhout†[2], Alessandro Zocca‡[2], Jeroen Pruyn§[3], and Rob van der Mei¶[1,2]

[1]Stochastics Department CWI, Science Park 123, 1098 XG, Amsterdam, The Netherlands
[2]Mathematics Department VU Amsterdam, Boelelaan 1111, 1081 HV, Amsterdam, The Netherlands
[3]Department of Maritime and Transport Technology, TU Delft, Leeghwaterstraat 2628 CD, Delft, The Netherlands

Abstract

The maritime industry must prepare for the energy transition from fossil fuels to sustainable alternatives. Making ships future-proof is necessary given their long lifetime, but it is also complex because the future fuel type is uncertain. Within this uncertainty, one typically overlooks pipe routing, although it is a crucial driver for design time and costs. Therefore, we propose a mathematical approach for modeling uncertainty in pipe routing with deterministic, stochastic, and robust optimization. All three models are based on state-of-the-art integer linear optimization models for the Stochastic Steiner Forest Problem and adjusted to the maritime domain using specific constraints for pipe routing. We compare the models using both artificial and realistic instances and show that considering uncertainty using stochastic optimization and robust optimization leads to cost reductions of up to 22% in our experiments.

Keywords— Pipe routing, Ship design, Robust optimization, Stochastic optimization, Energy transition, Stochastic Steiner forest

1 Introduction

The maritime industry is a significant contributor to global greenhouse gas emissions as it accounts for 2-3% of global carbon emissions [21]. Therefore, the International Maritime Organization (IMO) and the United Nations (UN) have made regulations and guidelines for the maritime industry. To stop global warming, emissions must be reduced as soon as possible. Approximately a 40% reduction per vessel is to be achieved by 2030 and even net zero for the fleet by 2050 [22].

These guidelines should stimulate the maritime industry to transition from fossil fuels to sustainable alternatives. However, ship owners optimize for the economic situation at the beginning of a ship's construction [32]. This means sustainability is often not taken into account. Yet, with a service life of 20 to 30 years at least [7], a ship should be profitable in all (economic) conditions encountered in that period. As a result, ships are currently already sub-optimal at delivery. To deal with this issue, we must consider future alternative fuels already in the design phase to prevent this sub-optimality.

Transitioning to a new fuel type would impact the engine, the fuel storage, and the piping between the two. The first two are extensively researched already [25, 39], yet piping is often overlooked. Pipe routing takes over 50% of the total detail-design labor hours [30]. Additionally, the labor costs yield 60% of the total costs of a ship [1, Section 1.2]. Hence, pipe routing greatly determines both design time and costs.

Furthermore, pipe routing constraints are heavily dependent on the corresponding fuel type [28]. Even though present studies fundamentally mention the same alternative fuels, the industry has not agreed on a single optimal future marine fuel [31]. They state that the future mix of fuels will depend on their expected price, availability, and suitability for the specific ship.

This work focuses on the mathematical optimization of pipe routing in ship design in light of the uncertainty created by the energy transition. The goal is to find a pipe routing that minimizes the costs of installing pipes when

*berend.markhorst@cwi.nl
†joost.berkhout@vu.nl
‡a.zocca@vu.nl
§j.f.j.pruyn@tudelft.nl
¶rob.van.der.mei@cwi.nl

building a ship and adjusting this routing in the future, also referred to as retrofit costs. To that end, we build upon and combine aspects of pipe routing and the mathematical optimization literature.

Related pipe routing literature The existing literature on pipe routing is described in several surveys, see [37, 1, 5]. The synthesis tables from [5, Section 5] give a detailed overview of the current state-of-the-art. According to this table, the pipe routing literature mainly describes deterministic models that do not take uncertainty into account. Even more, considering the uncertainty about the energy transition can reduce future retrofit costs, which has not been done in the literature so far [5, Section 6].

Related mathematical optimization literature The goal of ship pipe routing is connecting multiple points, such as engine room(s) and tank(s), using as little material as possible while following the rules described in [28]. Mathematically, this is close to a Steiner Tree Problem (STP), which is a well-studied problem in combinatorial optimization, see [26] for a recent overview. We represent a ship by a graph in which the vertices denote (engine) room(s) and tank(s), and the edges denote a connection between those vertices where we can install pipes. The objective is to connect a set of terminals (i.e., a subset of vertices) in a given graph using edges with minimal total costs. For example, a set of terminals may consist of an engine and multiple fuel tanks. STP is known to be NP-hard [8, p. 208-209], meaning that it is unlikely that there exists a scalable (polynomial-time) algorithm that can find the optimal solution to all STP instances.

The stochastic equivalent of the STP is called the Stochastic Steiner Tree Problem (SSTP), which considers two stages and a finite number of scenarios with the corresponding probabilities, terminal groups, and edge costs. In the first stage, it is unknown which scenario will occur in the second stage. The question is which edges to buy in the first stage and which (more expensive) edges to buy in the second stage. Approximation algorithms for the SSTP are described in [17, 16, 36, 13, 18], whereas an exact model that uses a two-stage branch-and-cut algorithm based on Benders' decomposition is discussed in [6]. For the 11th DIMACS challenge, a genetic algorithm has been developed and is discussed in [20]. Additionally, a comparison between different Integer Linear Optimization (ILO) models for the SSTP is made in [38]. Finally, a two-stage branch-and-cut algorithm based on a decomposed model is discussed in [27], whereas a new decomposition model is described in [24].

In this work, multiple groups of terminals (i.e., tanks and engine rooms) must be connected in ship pipe routing. Consequently, we consider a generalization of STP called the Steiner Forest Problem (SFP) [35], in which multiple terminal groups must be connected. As STP is NP-hard, also SFP is NP-hard [9]. For this problem, approximation algorithms are described in [2, 40, 10], whereas a greedy algorithm is discussed in [15] and a local-search algorithm in [12]. We mainly base our work on a study [34] that describes and compares different ILO models for the SFP. We have only found one work that studies the Stochastic Steiner Forest Problem (SSFP) with an approximation algorithm [14]. To the best of our knowledge, no paper in the literature describes an SSFP where the terminal groups from the first stage scenario must also be connected.

Our contributions This work is the first to introduce the 2-Stage Stochastic Steiner Forest Problem (2S-SSFP), which we base on the SSTP problem formulation from [6, 38, 24]. Our work differs from these three papers as

- we generalize from STP to SFP by using multiple terminal groups per scenario;
- we generalize the problem by requiring the terminal groups from the first stage scenario to be connected.

We study deterministic (DO), stochastic (SO), and robust optimization (RO) models adjusted with pipe routing constraints. For these models, we build upon [34], as this work provides models with state-of-the-art LO relaxation bounds for SFP, leading to enhanced optimization performance in practice. To study the scalability of the models and the relative gains of considering uncertainty, we use artificial instances and a realistic instance made in collaboration with maritime experts.

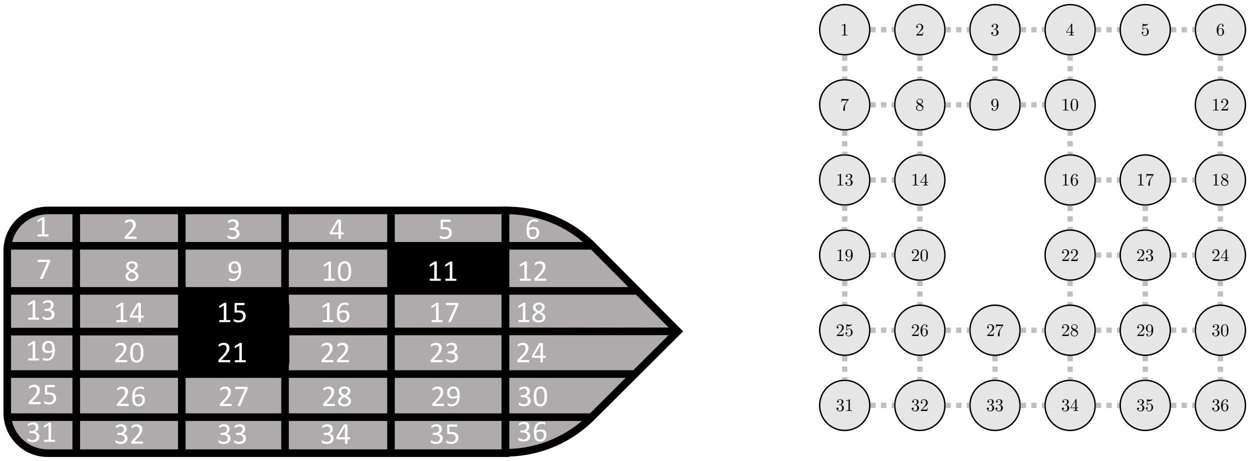
Outline The rest of this paper is structured as follows. Sections 2 and 3 formulate the deterministic and stochastic problem, respectively, and elaborate on the mathematical models. A discussion of the experiments and the corresponding results follows in Section 4. Finally, we conclude the paper with our conclusions and directions for future research in Section 5.

2 Deterministic problem formulation

In this section, we describe the deterministic problem formulation of the SFP. First, we give the general formulation of the SFP and explain how this general problem relates to ship pipe routing in Section 2.1. Then, we present the mathematical models of our deterministic optimization ILO model in Section 2.2.

2.1 Problem description for ship pipe routing

We model a ship as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V} := \{1, \dots, V\}$ denotes the set of (ship) rooms and $\mathcal{E}$ represents the possible pipe connections between them, i.e., $\mathcal{E} := \{(u, v) : u \in \mathcal{V}, v \in \mathcal{V}, u \sim v, u < v\}$, where $u \sim v$ denotes adjacency between the vertices u and v. Furthermore, $\mathcal{A}$ is the set of arcs of the bi-direction of G, represented by $\{(u, v) : u \in \mathcal{V}, v \in \mathcal{V}, u \sim v\}$. Each (ship) room can contain one or more engines or fuel tanks. We illustrate in Figure 1a an example of a schematic top-down view of a ship's deck and in Figure 1b the corresponding graph representation. Later, when we consider future scenarios, we will assume that the ship's graph representation G remains the same.



(a) **(b)**

Figure 1: Example of a ship pipe routing problem instance and the corresponding abstract representation as a graph. Figure 1a shows a schematic top-down view of a ship's deck. We can install pipes in the gray rooms but not in the black rooms. Figure 1b displays a graph representation of the ship's deck. Note that we omitted the three vertices whose corresponding rooms are located at the black squares of Figure 1a. Dashed lines denote possible pipe connections.

Pipes must connect each ship's engine to one or more tanks to enable fuel transportation. This information is captured by a given set of K terminal groups $\mathfrak{T} = (\mathcal{T}^k)_{k \in \mathcal{K}}$, where $\mathcal{T}^k \subseteq \mathcal{V}$ denotes the k-th terminal group consisting of terminal vertices, in short terminals, and $\mathcal{K} = \{1, \dots, K\}$. We must install pipes such that all vertices within a terminal group $\mathcal{T}^k$ are connected for all $k \in \mathcal{K}$. Without loss of generality, we can take these terminal groups to be pairwise disjoint subsets of vertices, i.e., $\bigcap_{k \in \mathcal{K}} \mathcal{T}^k = \emptyset$ [34, Section 1].

We refer to a scenario in this article as a pipe routing instance in the present or in the future (in contrast, the literature often denotes a scenario as a branch in the scenario tree, see Figure 3 for an example). We assume that every scenario (now and in the future) corresponds to one fuel type, which does not rule out the possibility that future scenarios differ from the present scenario. Different fuel types require different pipe types [28]. Therefore, we introduce a set of available pipes $\mathfrak{P} := \{1, \dots, P\}$ of size $P \in \mathbb{N}$. We also introduce the subset $\mathcal{P} \subseteq \mathfrak{P}$ that describes feasible pipes for the fuel type under consideration. Introducing $\mathfrak{P}$ and $\mathcal{P}$ allows us to consider different fuel types in the future. The cost of placing a pipe $p \in \mathfrak{P}$ at a particular edge $(u, v) \in \mathcal{E}$ is given $\gamma_{puv} > 0$ and all costs are collected in $\Gamma = (\gamma_{puv})_{p \in \mathfrak{P}, (u,v) \in \mathcal{E}}$.

Due to regulations, pipes transporting dangerous fuel types cannot cross certain rooms on a ship [28]. For example, diesel cannot be routed through a room adjacent to the waterside to reduce the chance of pollution in an accident. For this purpose, we introduce the set of admissible edges $\mathfrak{E} \subseteq \mathcal{E}$ for the fuel type under consideration, i.e., only edges in $\mathfrak{E}$ can be used to install pipes to connect terminal groups. Again, introducing both $\mathfrak{E}$ and $\mathcal{E}$ allows us to consider different fuel types in the future. Similarly, note that $\mathfrak{E} \subseteq \mathcal{E}$ induces a subset of admissible arcs $\mathfrak{A} \subseteq \mathcal{A}$, i.e., only arcs in $\mathfrak{A}$ can be used to connect vertices in terminal groups.

An instance of our problem is denoted by $I = (G, \mathfrak{P}, \mathfrak{T}, \mathcal{P}, \Gamma, \mathfrak{E})$, where G (and thus $\mathcal{A}$) and $\mathfrak{P}$ are fixed, and $\mathfrak{T}$, $\mathcal{P}$, Γ, and $\mathfrak{E}$ (and thus $\mathfrak{A}$) can vary in future scenarios. We define $\mathcal{M}(I)$ as the set of all feasible solutions for instance I. A feasible solution $\mathbf{S} \in \mathcal{M}(I)$ for a fuel type under consideration is given by a set of edge-pipe pairs that (i) connect all vertices in each terminal group from $\mathfrak{T}$, (ii) uses only feasible pipes from $\mathcal{P}$ for the terminal connections, and (iii) only uses feasible edges for the terminal connections. Note that a feasible $\mathbf{S} \in \mathcal{M}(I)$ may use different feasible pipes to connect terminals and can also install infeasible pipes as long as they are not used to connect terminals (this can be efficient to anticipate for future scenarios later on). To be able to take future scenarios into account, we assume that we are given the set $\mathbf{S}_0$ of edge-pipe pairs that describe which pipes already exist on which edge. Specifically,

$(p,(u,v)) \in \mathbf{S}_0$ means that pipe $p \in \mathcal{P}$ is present at edge $(u,v) \in \mathcal{E}$. We let $F(I, \mathbf{S}_0, \mathbf{S})$ be the cost of solution $\mathbf{S} \in \mathcal{M}(I)$ for instance I when edge-pipe pairs $\mathbf{S}_0$ are present. Then our mathematical optimization problem can be written as

$$\textbf{(DO)} \qquad \min_{\mathbf{S}\in\mathcal{M}(I)} F(I, \mathbf{S}_0, \mathbf{S}) = \min_{\mathbf{S}\in\mathcal{M}(I)} \sum_{(p,(u,v))\in\mathbf{S}\setminus\mathbf{S}_0} \gamma_{puv}. \tag{1}$$

Note that our problem, with $\mathbf{S}_0 = \emptyset$, reduces to a Steiner forest problem $(G, \mathfrak{T})$ if it holds that $\mathfrak{P} = \mathcal{P} = \{1\}$, $\mathcal{E} = \mathfrak{E}$ (and thus $\mathcal{A} = \mathfrak{A}$) in $I = (G, \mathfrak{P}, \mathfrak{T}, \mathcal{P}, \Gamma, \mathfrak{E})$, i.e., there is only one feasible pipe type, all edges are feasible, and there are no current pipes in the graph. Consequently, (DO) is NP-hard.

2.2 Deterministic ILO model for ship pipe routing

We can reformulate (DO) to an integer linear optimization (ILO) model so that (commercial) ILO solvers can solve our problem. This section introduces the deterministic ILO model and the corresponding decision variables and constraints. For the deterministic ILO model, we build on the work of [34], as this work describes the cut-based and flow-based ILO models for the SFP, which are equivalent. In this work, we use both the undirected and directed flow-based ILO models from [34] tailored to the ship pipe routing problem. An overview of the ILO models can be found in Table 1, which shows that we consider three optimization types (DO, RO, and SO) and two model types regarding the flow (undirected and directed). For example, (DO-U) and (DO-D) are ILO models for DO using undirected and directed formulations, respectively. We explain later the difference between the two model types.

Table 1: Abbreviations of our different ILO models. Model types specify whether directed flows or undirected flows are considered.

Optimization type \ model type	Undirected (U)	Directed (D)
Deterministic optimization (DO)	(DO-U)	(DO-D)
Robust optimization (RO)	(RO-U)	(RO-D)
Stochastic optimization (SO)	(SO-U)	(SO-D)

Deterministic optimization with an undirected formulation For (DO-U) in (2), we introduce a binary decision variable x_{puv} that equals 1 if pipe $p \in \mathfrak{P}$ is installed at edge $(u,v) \in \mathcal{E}$, and 0 else. To ensure that pipes connect all terminals in each terminal group, we will let an artificial flow go through the pipes. In particular, in each terminal group $k \in \mathcal{K}$, we randomly (without losing on generality) designate one terminal as the root vertex $r_k \in \mathcal{T}^k$ of that particular terminal group for orientation purposes. The set of all root vertices is defined as $\mathcal{R} = \{r_1, \dots r_K\}$. From the root vertex, we let an (artificial) flow go to all remaining terminals in the corresponding terminal group. We define f_{tpuv} as the flow amount to terminal $t \in \mathfrak{T} \setminus \mathcal{R}$ from the corresponding root over pipe $p \in \mathcal{P}$ at arc $(u,v) \in \mathfrak{A}$. The formulation of (DO-U) is called "undirected" as it does not force a flow direction upon the installed pipes for different terminals. Table 2 gives the comprehensive list of the sets, parameters, and decision variables used to formulate (DO-U) in (2).

Table 2: Notation overview for our ship pipe routing problem (DO) and its ILO model (DO-U).

Sets	
$\mathcal{V}$	Set of vertices indexed from 1 to V.
$\mathcal{E}$	Set of edges, represented by $\{(u,v) : u \in \mathcal{V}, v \in \mathcal{V}, u \sim v, u < v\}$.
$\mathfrak{E}$	Set of admissible edges; $\mathfrak{E} \subseteq \mathcal{E}$.
$\mathcal{A}$	Set of arcs, represented by $\{(u,v) : u \in \mathcal{V}, v \in \mathcal{V}, u \sim v\}$.
$\mathfrak{A}$	Set of admissible arcs; $\mathfrak{A} \subseteq \mathcal{A}$.
$\mathcal{K}$	Set of indices for the terminal groups, indexed from 1 to K.
$\mathfrak{T}$	Set of all terminal groups $\mathfrak{T} = (\mathcal{T}^k)_{k\in\mathcal{K}}$, where $T^k \subseteq \mathcal{V}$.
$\mathcal{R}$	Set of root vertices; $\mathcal{R} = \{r^1, \dots r^k\}$, where $r^k \in \mathcal{T}^k$ for terminal group $k \in \mathcal{K}$. Note that the root vertex is chosen arbitrarily for each terminal group.
$\mathfrak{P}$	Set of available pipe types, indexed from 1 to P.
$\mathcal{P}$	Set of feasible pipe types; $\mathcal{P} \subseteq \mathfrak{P}$.
Parameters	

γ_{puv}	Cost parameter for installing pipe type $p \in \mathfrak{P}$ at edge $(u,v) \in \mathcal{E}$.
Other notation	
$\tau(t)$	The index of the unique terminal group that contains the terminal $t \in \mathfrak{T}$.
Decision Variables	
x_{puv}	Binary variable equal to 1 if we install a pipe of type $p \in \mathfrak{P}$ on edge $(u,v) \in \mathcal{E}$, and 0 otherwise.
f_{tpuv}	Binary variable equal to 1 if there is a flow over arc $(u,v) \in \mathfrak{A}$ with pipe type $p \in \mathcal{P}$ for a route between vertex $t \in \mathfrak{T} \setminus \mathfrak{R}$ and $r^{\tau(t)}$, and 0 otherwise.

(DO-U)

$$\min \sum_{(p,(u,v)) \in (\mathfrak{P} \times \mathcal{E}) \setminus \mathbf{S}_0} x_{puv} \cdot \gamma_{puv} \tag{2a}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}} \left(\sum_{u:(v,u) \in \mathfrak{A}} f_{tpvu} - \sum_{u:(u,v) \in \mathfrak{A}} f_{tpuv} \right) = \begin{cases} 1 & \text{if } v = r^{\tau(t)} \\ -1 & \text{if } v = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall v \in \mathcal{V}, \forall t \in \mathfrak{T} \setminus \mathfrak{R} \tag{2b}$$

$$f_{tpuv} + f_{tpvu} \leq x_{puv} \qquad \forall t \in \mathfrak{T} \setminus \mathfrak{R}, \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{E} \tag{2c}$$

$$f_{tpuv} \in \mathbb{B} \qquad \forall t \in \mathfrak{T} \setminus \mathfrak{R}, \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{A} \tag{2d}$$

$$x_{puv} \in \mathbb{B} \qquad \forall p \in \mathfrak{P}, \forall (u,v) \in \mathcal{E} \tag{2e}$$

The goal of (2) is to minimize the costs of connecting the terminals within all terminal groups (2a) while ensuring the conservation of flows as formulated in (2b) using only admissible edges and feasible pipe types. Note that we added the summation over the set of pipes to allow for different (feasible) pipe types within a connection between a root and terminal. We ensure that an edge is used only in one direction for each connection between $r^{\tau(t)}$ and terminal t and connect the decision variables f_{tpuv} and x_{puv} in (2c). Note that we restrict to feasible pipe types and admissible edges in our flow decision variable f_{tpuv} and that (2e) can be relaxed because (2d) enforces the integrality of x_{puv}. For simplicity, and with a slight abuse of notation, we may capture all feasible $\mathbf{x}$ and $\mathbf{f}$ for (DO-U) in $\mathcal{M}(I)^{\text{(DO-U)}}$, so that (DO-U) can be compactly written as

$$\textbf{(DO-U)} \qquad \min \sum_{(p,(u,v)) \in (\mathfrak{P} \times \mathcal{E}) \setminus \mathbf{S}_0} x_{puv} \cdot \gamma_{puv} \tag{3a}$$

$$\text{s.t.} \qquad \mathbf{x}, \mathbf{f} \in \mathcal{M}(I)^{\text{(DO-U)}} \tag{3b}$$

Deterministic optimization with a directed formulation To solve an ILO model efficiently, it is important to have an ILO model that gives sharp LO-relaxation bounds [34]. The formulation of (DO-U) can be sharpened to give better LO-relaxation bounds. The LO-relaxation of (DO-U) now allows for directed cycles of flows of different terminals, which is not tight. These directed cycles can appear for different terminals from one terminal group or when two (or more) terminal groups are connected/overlapping in the solution. By ruling out these directed cycles of flow in the LO-relaxation, a stronger model is obtained. See Figure 2b for an illustrative example.

To rule out the directed cycles of flow, we introduce an ILO model (DO-D) that "directs" the flow into consistent orientations. More specifically, (DO-D) improves over (DO-U) by 1) dynamically combining overlapping terminal groups effectively into one large terminal group and 2) finding an arborescence (a directed tree) for this combined terminal group in which one root is designated to send (artificial) flow to all other terminals in the combined terminal group. In (DO-D), it is ensured that all flow in an arborescence has the same orientation in the LO-relaxation. As a result, directed cycles are eliminated, as shown in Figure 2c. Hence, the LO-relaxation from (DO-D) will be tighter than the LO-relaxation from (DO-U). More details can be found in [34].

For (DO-D), the decision variable x_{puv} from (DO-U) remains the same. Compared to (DO-U), the flow decision variable is extended by a terminal group index $k \in \mathcal{K}$ and the resulting f_{ktpuv} is the flow sent from the root of terminal group k to terminal t via pipe p at arc (u,v). Binary decision variable z_{kl} is 1 when the root of the terminal group k sends flow to all terminals of the terminal group l, and 0 else. When $z_{kl} = 1$, binary decision variable y_{kpuv} equals 1 when flow from the root of terminal group k is sent over arc (u,v) through pipe p, and 0 else. Lastly, binary decision

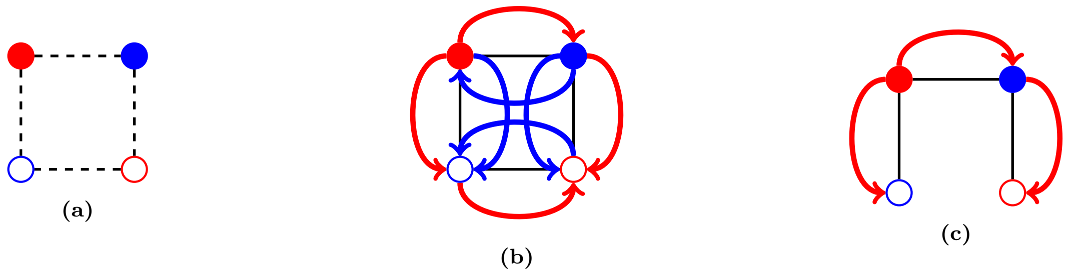


Figure 2: Example that shows the difference between the (DO-U) and (DO-D). Figure 2a shows a small example from Figure 1 and 2 of [34] with two terminal groups, red and blue, and possible connections between vertices denoted by dashed lines. The roots are denoted by circles that are filled with colors. The remaining vertices are the terminals. The costs of using an edge equal 1 for all edges in the graph. The optimal integer solution yields an objective of 3 because it requires three edges to connect all terminals. Figure 2b shows a feasible fractional solution for (DO-U) where the colored arcs denote flows of 0.5 for the corresponding root-terminal pairs, and the corresponding $\mathbf{x}$ decision variables equal 0.5 and are represented by the solid black lines. This solution is infeasible for (DO-D) because there are directed cycles between the two roots and the two terminals. Figure 2c shows a feasible solution for (DO-D) where the $\mathbf{x}$ and $\mathbf{f}$ decision variables both equal 1. The red arcs represent the arborescence that connects both terminal groups, which yields a solution with three edges.

variable y_{puv} equals 1 when pipe p at arc (u, v) is used to send flow over by any of the created arborescences, and 0 else. An overview of all decision variables is given in Table 3, which builds upon the notation from Table 2.

Table 3: Notation overview for the (DO-D) problem formulation and ILO model, which builds on the notation from Table 2.

Sets	
$\mathfrak{R}$	Set of non-terminal vertices, also called Steiner vertices: $\mathfrak{R} = \mathcal{V} \setminus \mathfrak{T}$.
$\mathfrak{T}^{k\dots K}$	Set of some terminal groups: $\mathfrak{T}^{k\dots K} = (\mathcal{T}^k)_{k\in\{k,\dots,\mathcal{K}\}}$.
$\mathfrak{T}_r^{k\dots K}$	Set of some terminal groups without the corresponding root vertex: $\mathfrak{T}_r^{k\dots K} = \mathfrak{T}^{k\dots K} \setminus \{r^k\}$.
Decision Variables	
f_{ktpuv}	Binary variable equal to 1 if there is a flow over arc $(u, v) \in \mathfrak{A}$ with pipe type $p \in \mathcal{P}$ for a route between root r^k of terminal group $k \in \mathcal{K}$ and terminal $t \in \mathfrak{T}_r^{k\dots K}$, and 0 otherwise.
y_{kpuv}	Binary variable equal to 1 if there is a flow over arc $(u, v) \in \mathfrak{A}$ with pipe of type $p \in \mathcal{P}$ from root r^k of terminal group $k \in \mathcal{K}$, and 0 otherwise.
y_{puv}	Binary variable equal to 1 if there is a flow over arc $(u, v) \in \mathfrak{A}$ with pipe of type $p \in \mathcal{P}$, and 0 otherwise.
z_{kl}	Binary variable equal to 1 if $\mathcal{T}^k$ and $\mathcal{T}^l$ are in the same arborescence and r^k is the root for both terminal groups, and 0 otherwise.

Using the extra notation, we get the following ILO model (DO-D):

(DO-D)

$$\min \sum_{(p,(u,v))\in(\mathfrak{P}\times\mathcal{E})\setminus\mathbf{S}_0} x_{puv} \cdot \gamma_{puv} \tag{4a}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}} \Bigg(\sum_{u:(v,u)\in\mathfrak{A}} f_{ktpvu} - \sum_{u:(u,v)\in\mathfrak{A}} f_{ktpuv} \Bigg) = \begin{cases} z_{kl} & \text{if } v = r^k \\ -z_{kl} & \text{if } v = t \\ 0 & \text{otherwise} \end{cases} \qquad \begin{cases} \forall k \in \mathcal{K}, \forall t \in \mathfrak{T}_r^{k\dots K} \\ \forall v \in \mathcal{V} \text{ with } \tau(t) = l \end{cases} \tag{4b}$$

$$f_{ktpuv} \le y_{kpuv} \qquad \begin{cases} \forall k \in \mathcal{K}, \forall t \in \mathfrak{T}_r^{k\dots K} \\ \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{A} \end{cases} \tag{4c}$$

$$\sum_{k\in\mathcal{K}} y_{kpuv} \le y_{puv} \qquad \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{A} \tag{4d}$$

$$y_{puv} + y_{pvu} \le x_{puv} \qquad \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{E} \tag{4e}$$

$$\sum_{l=1}^{k} z_{lk} = 1 \qquad \forall k \in \mathcal{K} \tag{4f}$$

$$z_{kk} \geq z_{kl} \qquad \begin{cases} \forall k \in \mathcal{K} \setminus \{1, K\} \\ \forall l \in \mathcal{K} \text{ if } l \geq k+1 \end{cases} \tag{4g}$$

$$\sum_{p \in \mathcal{P}} \sum_{u:(u,v) \in \mathfrak{A}} y_{puv} \leq 1 \qquad \forall v \in \mathcal{V} \tag{4h}$$

$$\sum_{p \in \mathcal{P}} \sum_{u:(u,t) \in \mathfrak{A}} y_{kpuv} = 0 \qquad \forall k \in \mathcal{K} \setminus \{1\}, \forall t \in \mathfrak{T}^{1 \ldots k-1} \tag{4i}$$

$$\sum_{p \in \mathcal{P}} \sum_{u:(t,u) \in \mathfrak{A}} f_{ktpuv} = 0 \qquad \forall k \in \mathcal{K}, \forall t \in \mathfrak{T}_r^{k \ldots K} \tag{4j}$$

$$\sum_{p \in \mathcal{P}} \sum_{u:(u,v) \in \mathfrak{A}} y_{puv} \leq \sum_{p \in \mathcal{P}} \sum_{u:(v,u) \in \mathcal{A}} y_{puv} \qquad \forall v \in \mathfrak{R} \tag{4k}$$

$$\sum_{p \in \mathcal{P}} \sum_{u:(u,v) \in \mathfrak{A}} y_{kpuv} \leq \sum_{p \in \mathcal{P}} \sum_{u:(v,u) \in \mathcal{A}} y_{kpuv} \qquad \forall k \in \mathcal{K}, \forall v \in \mathcal{V} \setminus \mathfrak{T}_r^{k \ldots K} \tag{4l}$$

$$\sum_{u:(u,r^l) \in \mathfrak{A}} y_{kpur^l} \leq z_{kl} \qquad \begin{cases} \forall k \in \mathcal{K} \setminus K \\ \forall l \in \mathcal{K} \text{ if } l \geq k+1 \\ \forall p \in \mathcal{P}. \end{cases} \tag{4m}$$

$$f_{ktpuv} \in \mathbb{B} \qquad \begin{cases} \forall k \in \mathcal{K}, \forall t \in \mathfrak{T}_r^{k \ldots K} \\ \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{A} \end{cases} \tag{4n}$$

$$x_{puv} \in \mathbb{B} \qquad \forall p \in \mathfrak{P}, \forall (u,v) \in \mathcal{E} \tag{4o}$$

$$y_{puv} \in \mathbb{B} \qquad \forall p \in \mathcal{P}, \forall (u,v) \in \mathfrak{A} \tag{4p}$$

$$y_{kpuv} \in \mathbb{B} \qquad \begin{cases} \forall k \in \mathcal{K}, \forall p \in \mathcal{P} \\ \quad \forall (u,v) \in \mathfrak{A} \end{cases} \tag{4q}$$

$$z_{kl} \in \mathbb{B} \qquad \forall k \in \mathcal{K}, \forall j \in \{k \ldots K\} \tag{4r}$$

The objective (4a) is similar to (2a), whereas the conservation of flows in (4b) differs from (2b). This constraint ensures that each terminal is contained in an arborescence rooted at r^k for some $k \in \mathcal{K}$. From each arborescence root r^k, an artificial flow is sent to all remaining terminals in the arborescence. Decision variables f_{ktpuv} activate y_{kpuv} in (4c) whenever flow is sent from root r^k to terminal t via pipe p at arc (u,v). We ensure that every arc is part of at most one arborescence in (4d). In case of an overlap, the corresponding arborescences are forced to be merged into one arborescence. Similarly to (2c), (4e) allows for only one direction on an edge. We enforce that every terminal group is rooted at exactly one root in (4f), whereas (4g) enforces exactly one root in each arborescence.

Constraints (4h)-(4m) are not necessary for (DO-D) to produce feasible solutions but are introduced to strengthen the model's LO-relaxation [34]. In (4h), we ensure that every vertex receives flow over at most one pipe. According to the definition of z_{kl}, the root r^k can be only responsible for terminal groups $l \geq k$. Consequently, (4i) prevents a connection between root r^k and a terminal from $\mathfrak{T}^{1 \ldots k-1}$. Constraints (4j) prevent a flow from leaving a terminal. We denote flow-balance constraints in (4k) and (4l), also mentioned in [24, Section 2.2] for the SSTP, which state that the in-degree of a Steiner vertex cannot be larger than its out-degree: (4k) enforces this in the overall solution whereas (4l) focuses on each terminal group. We enforce that the arborescence rooted at r^k can use root r^l if and only if $z_{kl} = 1$ in (4m).

Finally, we include integrality constraints in (4n)-(4r). As (4n) already enforces integrality on y_{kpuv}, constraints (4p) and (4q) can be relaxed.

For simplicity, and with a slight abuse of notation, we may capture all feasible $\mathbf{x}$, $\mathbf{f}$, $\mathbf{y}$, and $\mathbf{z}$ for (DO-D) in $\mathcal{M}(I)^{\text{(DO-D)}}$, so that (DO-D) can be written as

$$\textbf{(DO-D)} \qquad \min \sum_{(p,(u,v)) \in (\mathfrak{P} \times \mathcal{E}) \setminus \mathbf{S}_0} x_{puv} \cdot \gamma_{puv} \tag{5a}$$

$$\text{s.t.} \qquad \mathbf{x}, \mathbf{f}, \mathbf{y}, \mathbf{z} \in \mathcal{M}(I)^{\text{(DO-D)}} \tag{5b}$$

3 Accounting for uncertainty: two new problem formulations

In this section, we explain how uncertainty affects the ship pipe routing problem and introduce two new optimization models to deal with it. Stochastic optimization (SO) and robust optimization (RO) are techniques that address uncertainties and variability in real-world optimization problems. They are different approaches for dealing with uncertainty; SO requires more detailed distributional information and focuses on the average case, whereas RO requires information on the support of the uncertain parameters and focuses on the worst case. Although both models are well known in the mathematical literature, applying these models in our practical context is novel [5, Section 6]. For more details about SO, also known as Stochastic Programming in the literature, we refer to [4, 23], whereas [3, 11] provide more information on RO.

3.1 Benefits of using SO and RO for pipe routing

Currently, diesel is the most used ship fuel [31]. As part of the energy transition, guidelines from the IMO [22] stimulate the maritime industry to transition to alternative, less polluting fuels. However, no single alternative fuel is recommended to be used in the future. The future mix of fuels used depends on many external factors, such as technology improvements, availability, and future costs [31]. In our framework, we consider two periods in time, the present and the future, and refer to them as the first stage and second stage, respectively. Figure 3 shows the scenario tree schematically representing our problem setting. At each stage, pipe routes can be changed by installing extra pipes. We assume that it is not necessary to remove unused pipes. In the first stage, we consider one scenario, typically diesel. However, in the second stage, we consider multiple scenarios, each denoting a different fuel type and possibly different pipe routes. For example, a future scenario could be methanol, which, unlike diesel, can be routed through rooms next to the waterside and requires double-walled pipes [28]. Because of these different characteristics, methanol might need different pipes and routes than diesel.

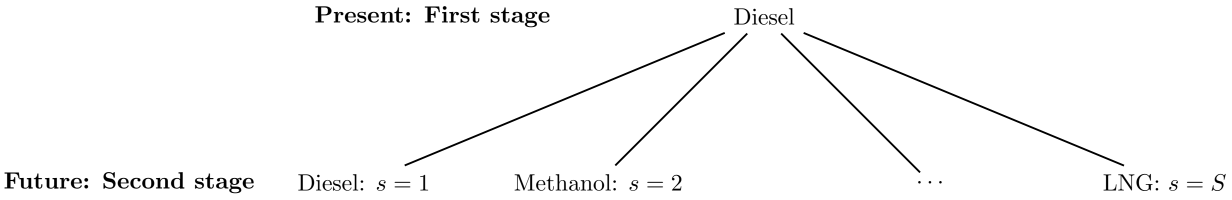


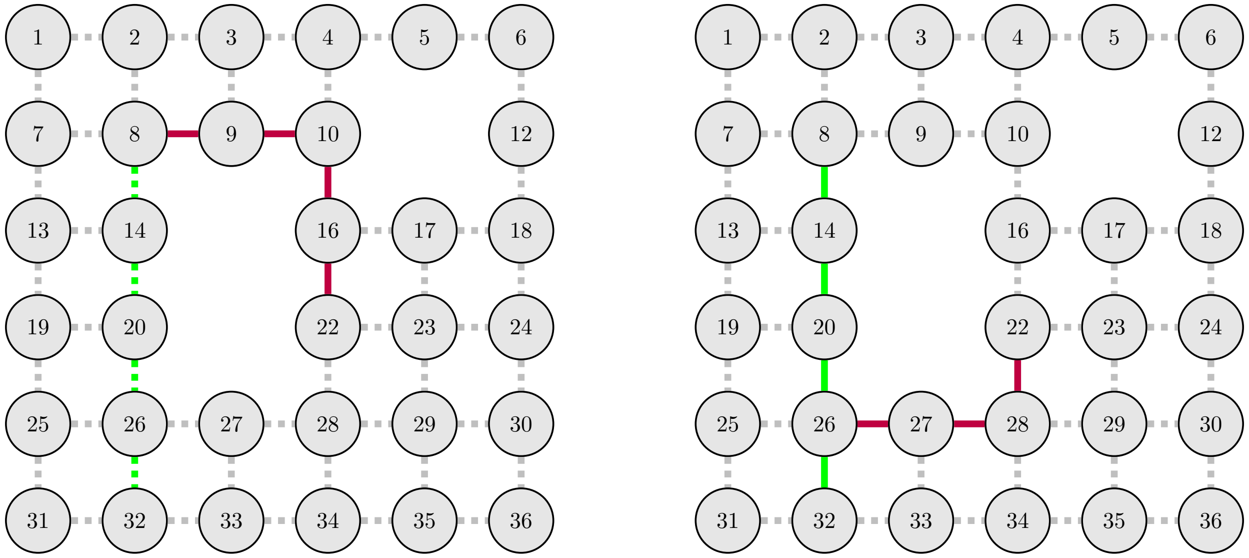
Figure 3: Two-stage scenario tree that schematically represents the problem setting we are studying.

Ship designers optimize for the economic situation of a ship at the start of construction [32, 33]. As a result, ships are already sub-optimal at delivery due to ongoing technological developments. Changing the ship's pipe network after construction is not preferable. For example, retrofitting a ship from diesel to methanol is expensive because of the ship's downtime and the required complex maintenance. To make ships future-proof for both economic and sustainability reasons, alternative fuels must be taken into account in the design phase.

To illustrate the importance of considering future fuels, we use the example from Figure 1b. The first two branches from the scenario tree in Figure 3 are considered in this example: transitioning from diesel to diesel or methanol, respectively. For both fuel types, we need to connect the fuel tanks and the engine room with a pipe route. More specifically, vertices 22 and 32 in Figure 1b represent different fuel tanks, diesel, and methanol, respectively, whereas vertex 8 denotes the engine room. Figures 4a and 4b show a deterministic and robust pipe route for this example, respectively. Because Figure 4a represents the deterministic solution, the pipe route is only optimal for the first stage, in which the diesel tank gets connected to the engine room. However, it is unknown which fuel type will be used in the future and therefore which scenario will be realized. If diesel persists in the future, the pipe route will not need to change. However, in case methanol occurs in the future, we need to build an extra pipe route in order to connect the methanol tank and the engine room, which is denoted by the green dashed line. Depending on the retrofit costs and the probabilities of the second scenario, a different pipe routing in the first stage can be optimal. For example, Figure 4b shows a robust solution that does not need any installations in the future. Although this route seems inefficient for the diesel scenario, it prepares the ship for a possible transition to methanol in the future. The only way to make the best pipe routing decisions is to include uncertainty in our mathematical models explicitly.

3.2 Extension of the deterministic problem formulation to SO and RO

This section extends the deterministic pipe routing problem formulations to SO and RO formulations. As mentioned in Section 2.1, we need to introduce a set of second-stage scenarios $s \in \mathcal{S}$, where each scenario corresponds to one fuel type. We reuse all previously introduced notations for (DO-U) and (DO-D) but then for each future scenario, which will be indicated by superscript (s) for scenario s. For example, $I^{(s)} = (G, \mathfrak{P}, \mathfrak{T}^{(s)}, \mathcal{P}^{(s)}, \Gamma^{(s)}, \mathfrak{E}^{(s)})$ denotes a problem instance in scenario s (in which $\mathfrak{T}^{(s)}$ for example denotes all the corresponding terminal groups) and $\mathbf{S}^{(s)}$ captures the decisions taken in scenario s. The cost of installing a pipe $p \in \mathfrak{P}$ at a particular edge $(u, v) \in \mathcal{E}$ is given



(a) Deterministic pipe route.

(b) Robust pipe route.

Figure 4: Example of a deterministic and robust pipe route connecting fuel tanks (vertex 22 for diesel and vertex 32 for methanol) with the engine room (vertex 8). The purple and green lines represent single and double-walled pipes, respectively. The solid lines denote pipes installed in the first stage, whereas the dashed lines denote pipes installed in the second stage if the methanol scenario takes place. The gray dashed lines represent the remaining edges.

by $\gamma^{(s)}_{puv} = \gamma_{puv} \cdot \lambda^{(s)}$ where $\lambda^{(s)} > 1$ is the inflation rate. These costs are captured in $\Gamma^{(s)} = (\gamma^{(s)}_{puv})_{p\in\mathfrak{P},(u,v)\in\mathcal{E},s\in\mathcal{S}}$. For an overview of the notation for the new problem formulations, see Table 4.

Table 4: Notation overview for (RO), (SO) and their ILO models, which builds on the notation from Table 2 and 3.

Sets	
$\mathcal{S}$	Set of scenario indices, indexed from 1 to S.
Decision Variables	
d	Continuous variable that captures the worst-case retrofit costs for the (RO-U) and (RO-D) models.

Using the new notation, we get the following robust optimization problem in which we anticipate the worst-case future scenario:

(RO)

$$\min_{\mathbf{S}\in\mathcal{M}(I)} \left(F(I,\emptyset,\mathbf{S}) + \max_{s\in\mathcal{S}} \min_{\mathbf{S}^{(s)}\in\mathcal{M}(I^{(s)})} \left(F(I^{(s)},\mathbf{S},\mathbf{S}^{(s)}) \right) \right) \tag{6a}$$

$$= \min_{\mathbf{S}\in\mathcal{M}(I)} \Bigg(\underbrace{\sum_{(p,(u,v))\in\mathbf{S}\setminus\mathbf{S}_0} \gamma_{puv}}_{\text{First stage costs}} + \underbrace{\max_{s\in\mathcal{S}} \min_{\mathbf{S}^{(s)}\in\mathcal{M}(I^{(s)})} \sum_{(p,(u,v))\in\mathbf{S}^{(s)}\setminus\mathbf{S}} \gamma^{(s)}_{puv}}_{\text{Second stage costs}} \Bigg). \tag{6b}$$

We can rewrite (RO) to an undirected ILO model denoted by (RO-U) that (commercial) ILO solvers can solve:

(RO-U)

$$\min \sum_{(p,(u,v))\in(\mathcal{P}\times\mathfrak{E})} x_{puv}\cdot\gamma_{puv} + d \tag{7a}$$

$$\text{s.t.}\quad d \geq \sum_{(p,(u,v))\in(\mathcal{P}^{(s)}\times\mathfrak{E}^{(s)})} (x^{(s)}_{puv} - x_{puv})\cdot\gamma^{(s)}_{puv} \qquad \forall s\in\mathcal{S} \tag{7b}$$

$$x^{(s)}_{puv} \geq x_{puv} \tag{7c}$$

$$\mathbf{x}, \mathbf{f} \in \mathcal{M}(I)^{(\text{DO-U})} \tag{7d}$$

$$\mathbf{x}^{(s)}, \mathbf{f}^{(s)} \in \mathcal{M}(I^{(s)})^{(\text{DO-U})}, \tag{7e}$$

where continuous decision variable d captures the worst-case retrofit costs. We force $x^{(s)}_{puv} = 1$ when $x_{puv} = 1$ in (7c), as we can reuse pipe $p \in \mathfrak{P}$ at edge $(u, v) \in \mathcal{E}$ in scenario $s \in \mathcal{S}$. We can also rewrite (RO) to a directed ILO model (RO-D) that is based on (DO-D). So (RO-D) is the same as (RO-U) above, but with $\mathcal{M}(I)^{(\text{DO-U})}$ replaced by $\mathcal{M}(I)^{(\text{DO-D})}$.

The stochastic optimization problem (SO), which aims to minimize the sum of expected costs, can be written as:

(SO)

$$\min_{\mathbf{S}\in\mathcal{M}(I)} \Big(F(I, \emptyset, \mathbf{S}) + \mathbb{E}_{\mathbb{S}}\Big[\min_{\mathbf{S}'\in\mathcal{M}(I^{\mathbb{S}})} \Big(F(I^{\mathbb{S}}, \mathbf{S}, \mathbf{S}')\Big)\Big]\Big) \tag{8a}$$

$$= \min_{\mathbf{S}\in\mathcal{M}(I)} \Big(\underbrace{\sum_{(p,(u,v))\in\mathbf{S}\setminus\mathbf{S}_0} \gamma_{puv}}_{\text{First stage costs}} + \underbrace{\mathbb{E}_{\mathbb{S}}\Big[\min_{\mathbf{S}'\in\mathcal{M}(I^{\mathbb{S}})} \sum_{(p,(u,v))\in\mathbf{S}^{(s)}\setminus\mathbf{S}} \gamma^{(s)}_{puv}\Big]}_{\text{Second stage costs}}\Big), \tag{8b}$$

where $\mathbb{S}$ denotes a discrete random variable for the future scenario with a known distribution, i.e., $\Pr(\mathbb{S} = s)$ is known for all $s \in \mathcal{S}$.

We can rewrite (SO) to an undirected ILO model denoted by (SO-U) that (commercial) ILO solvers can solve:

(SO-U)

$$\min \sum_{(p,(u,v))\in(\mathcal{P}\times\mathfrak{E})} x_{puv} \cdot \gamma_{puv} + \sum_{s\in\mathcal{S}} \rho^{(s)} \sum_{(p,(u,v))\in(\mathcal{P}^{(s)}\times\mathfrak{E}^{(s)})} (x^{(s)}_{puv} - x_{puv}) \cdot \gamma^{(s)}_{puv} \tag{9a}$$

$$\text{s.t.} \quad x^{(s)}_{puv} \geq x_{puv} \tag{9b}$$

$$\mathbf{x}, \mathbf{f} \in \mathcal{M}(I)^{(\text{DO-U})} \tag{9c}$$

$$\mathbf{x}^{(s)}, \mathbf{f}^{(s)} \in \mathcal{M}(I^{(s)})^{(\text{DO-U})}. \tag{9d}$$

We can also write (SO) to a directed ILO model (SO-D) that is based on (DO-D). So (SO-D) is the same as (SO-U), but with $\mathcal{M}(I)^{(\text{DO-U})}$ replaced by $\mathcal{M}(I)^{(\text{DO-D})}$.

4 Results

To highlight different aspects of the 2S-SSFP and the proposed models, we consider three graphs of different sizes and with different levels of modeling realism. This section will discuss our experiments on these graphs and report the corresponding results. Section 4.1 elaborates on the small graph from Figure 1b, whereas Section 4.2 and Section 4.3 discuss the artificial and realistic graphs, respectively. We compare the scalability of the models and study the relative gains of taking uncertainty into account. Both (DO-U) (2) and (DO-D) (4) are deterministic models that do not consider the uncertainty of the second stage and only optimize for the first stage, which is a naive approach compared to SO and RO. In the following, DO will mainly serve as a benchmark to compare with SO and RO. All experiments are executed on a supercomputer with 32 cores and CPU 2.4GHz using the Gurobi solver [19] for our Python code, which is available online.

4.1 Small graph: an example of the added value from SO

This section will discuss an example that shows the added value from SO and gives insight into the cost difference between the three mathematical models. We use the small and simple graph from Figure 1b, which is a grid graph where the costs of installing a single-walled pipe in the first stage equals 1, i.e.,

$$\gamma_{1uv} = 1, \qquad \forall (u, v) \in \mathcal{E}.$$

We assume that double-walled pipes are more expensive than single-walled ones for all three graphs. More specifically, we set the cost ratio between double-walled and single-walled pipes equal to 2, i.e.,

$$\gamma_{2uv} = 2\gamma_{1uv}, \qquad \forall (u, v) \in \mathcal{E}.$$

We assume that the costs of installing pipes are higher in the future than in the present, but we do not know to what extent. Therefore, we use the smallest integer that is greater than 1 for the ratio between future and present costs of installing pipes, i.e.,

$$\frac{\gamma_{puv}^{(s)}}{\gamma_{puv}} = \lambda^{(s)} = 2, \qquad \forall s \in \mathcal{S}, \forall p \in \mathcal{P}, \forall (u,v) \in \mathcal{E}.$$

Here, $\lambda^{(s)} = 2$ is a constant increase rate for the costs of installing pipes.

In the example from Figure 1b, we start with a diesel scenario in the first stage and transition to either a diesel or methanol scenario in the second stage. In the following, ρ_2 denotes the probability of the methanol scenario in the future, which consists of a transition from diesel to methanol. Since there are only two scenarios in this example, the likelihood of the first scenario is also known: $\rho_1 = 1 - \rho_2$. We can algebraically compute the DO, RO, and SO costs for this example. There are three pipe routes for the problem instance in Figure 1b. The first two routes are DO and RO, as shown in Figures 4a and 4b. DO installs four single-walled pipes in the first stage and possibly four double-walled pipes in the second stage, whereas RO installs three single-walled pipes and four double-walled pipes in the first stage. Figure 5a shows the third route, which installs three single-walled and three double-walled pipes in the first stage and one double-walled pipe in the second stage in the case of the methanol scenario. This route saves costs in the first stage but still prepares for a possible transition in the second stage. As mentioned, the second stage costs of installing a pipe are twice as high compared to the first stage. For a fair comparison, we also consider the expected second-stage costs of installing pipes for DO, which we compute by multiplying the second-stage costs by the probability of the methanol scenario. As a result, the expected DO costs increase linearly with ρ_2. The sum of the expected first and second-stage costs for the three routes are:

$$\begin{aligned}
&\mathbb{E}R_1(\rho_2) = 4 + 16\rho_2\\
&\mathbb{E}R_2(\rho_2) = 11\\
&\mathbb{E}R_3(\rho_2) = 9 + 4\rho_2,
\end{aligned}$$

where R_1 and R_2 denote the objective values for the DO and RO solution, and R_3 represents the objective value for the solution from Figure 5a. Note that both $\lambda^{(s)}$ and ρ_2 determine the slope of R_1 and R_3. Figure 5b visualizes the expected costs, where the x-axis and the y-axis show ρ_2 and the sum of the expected first- and second-stage costs, respectively. RO yields a horizontal line because this model does not depend on probabilities.

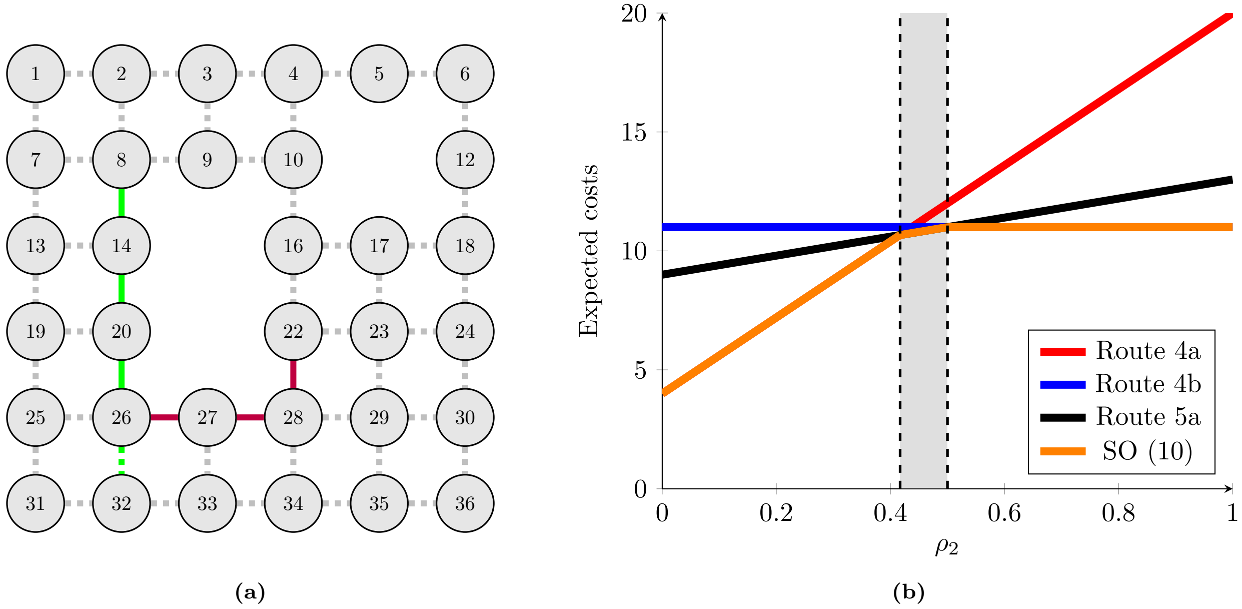


Figure 5: Figure 5a shows the third pipe route option for the example of Figure 1b. Note the similarity with Figure 4b, except the dashed line between vertices 26 and 32, which denotes that we possibly install a pipe there in the second stage. Figure 5b shows the expected costs of the three routes and (10) in Figure 1b. The shaded area indicates the interval of ρ_2 values for which $\mathbb{E}R_{\text{SO}}(p_2) \leq \min\{\mathbb{E}R_1(p_2), \mathbb{E}R_2(p_2)\}$.

SO takes the minimum of all three options:

$$\mathbb{E}R_{\text{SO}}(\rho_2) = \min\{4 + 16\rho_2, 11, 9 + 4\rho_2\}. \tag{10}$$

$\mathbb{E}R_3$ intersects with $\mathbb{E}R_1$ at $\rho_2 = \frac{5}{12}$ and with $\mathbb{E}R_2$ at $\rho_2 = \frac{1}{2}$. Between these two values, it holds that $\mathbb{E}R_{\mathrm{SP}} < \min\{\mathbb{E}R_1, \mathbb{E}R_2\}$, which shows the value of taking uncertainty into account. However, this result is based on the assumption that ρ_2 is known, which is not always the case in practice. Another way to quantify the added value of SO is the value of the stochastic solution (VSS), which is the difference between the expectation of the expected value solution (EEVS) and the optimal objective value of the SO. In this example, VSS equals the difference between $\mathbb{E}R_1$ and $\mathbb{E}R_{\mathrm{SO}}$, which ranges between 0 and 9 (82% of the SO objective). In other words, the relative gains of SO are relatively high in this example.

4.2 Artificial instance: a comparison of the models

In this section, we compare the scalability of the three models and show the relative gains of considering uncertainty. To this end, we introduce an artificial graph, which is a 5-by-5 grid graph. The costs of installing a single-walled pipe are randomly drawn from a uniform distribution between 1 and 10, i.e.,

$$\gamma_{1uv} \sim \mathcal{U}[1, 10], \qquad \forall (u,v) \in \mathcal{E}.$$

The installation costs of double-walled pipes are again $\gamma_{2uv} = 2\gamma_{1uv}$, $\forall (u,v) \in \mathcal{E}$. The terminals are randomly chosen from all vertices. For a parameter study, we vary the number of scenarios between two and four, the number of terminals between three and five, and the number of terminal groups between one and three, which amounts to 27 different parameter settings in total for the artificial graph. To account for randomness, we generate 100 instances for each parameter setting, resulting in 2700 different instances for the same artificial graph. For the sake of simplicity, we assume that every terminal group can install all pipe types on every edge of the graph and that each scenario is equally probable, i.e., $\rho_s = \frac{1}{S}, \forall s \in \mathcal{S}$.

Figure 6a shows the average compilation and run time for (SO-U) and (SO-D) over all the 2700 instances, respectively. The figure shows that directed formulations require more compilation time but yield a considerably shorter run time. We zoom in on this statement with Figure 6b, which displays the run times of (SO-U) and (SO-D) with the total number of terminals per scenario (i.e., the number of terminal groups multiplied by the number of terminals per terminal group). We see that (SO-D) is generally faster than (SO-U), typically when the number of terminals is large.

It is interesting to zoom in on the worst- and best-case instances in terms of run time for (SO-U). The best-case instance contains two scenarios, one terminal group per scenario, and three terminals per terminal group. These terminals lie close to each other, resulting in shorter run times. The worst-case instance contains more terminals, terminal groups, and scenarios, which leads to longer run times, probably due to an increased number of decision variables and constraints. When the terminals are spread across the whole graph, the routing problem becomes more complex, resulting in longer run times.

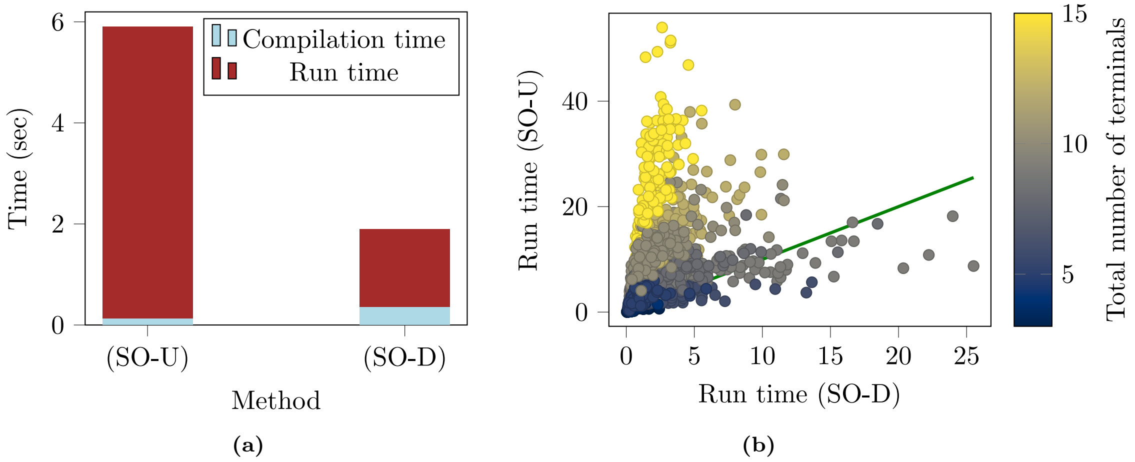


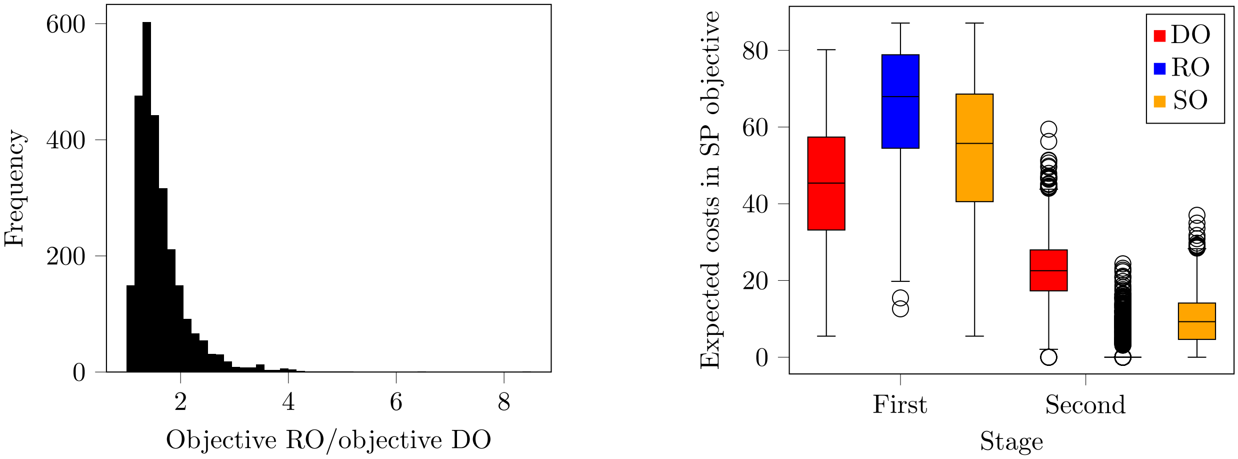
Figure 6: Figure 6a shows the average of the compilation and run times from (SO-U) and (SO-D). Figure 6b shows the run times of (SO-U) and (SO-D) with the total number of terminals per scenario (i.e., the number of terminal groups multiplied by the number of terminals per terminal group). The green line indicates where the run times of (SO-U) and (SO-D) are equal.

Because DO, RO, and SO are entirely different models, comparing them with one measure is not trivial. Therefore, we compare them to each other's objective in Table 5. In the DO objective, we divide the first-stage costs of the three models by the first-stage costs of DO. We see that RO yields a higher ratio than SO, which is caused by the conservatism of RO. SO and RO are 29% and 62% more expensive than DO in the DO objective, respectively. In the RO objective, we compute the objective value of the three solutions in case of the worst-case scenario and divide it by the RO objective value. We find that the objectives of SO and RO are closer to each other than those of DO and RO. Compared to RO, DO and SO are 29% and 9% more expensive, respectively. Hence, considering the uncertainty for the worst case yields a relative gain of 22%. Finally, in the SO objective, we compute the objective value of the three solutions in the average case with equal probabilities for each scenario and divide it by the SO objective value. DO and RO perform comparably in this objective as both lie approximately 5% from the optimal objective value. In other words, the VSS amounts to approximately 5%, which is the expected gain from solving the SO instead of DO.

Model \ objective	DO	RO	SO
DO	1.000	1.286	1.057
RO	1.623	1.000	1.048
SO	1.225	1.116	1.000

Table 5: Comparison of the models' average relative performances for three different objectives in artificial instances.

The following elaborates on the first- and second-stage costs of all three models. We compare the first-stage costs of DO and RO in Figure 7a. We divide the RO costs by the DO costs and find a right-skewed distribution, where the lower bound of this ratio yields 1. Figure 7b shows how the models perform in the first and second stages. Note that we compute the expected costs in the SO objective, which means that we compute the second stage costs for DO and RO in case of equal probabilities for each scenario, i.e., $\rho_1 = \rho_2 = 0.5$. We see that DO has the lowest first-stage costs but is relatively expensive in the second stage, whereas both SO and RO require a bigger investment in the first stage but have considerably lower second-stage costs. When the terminals are close to each other in the first stage but far apart in the second stage, DO seems affordable in the first stage and yields high costs in the second stage.



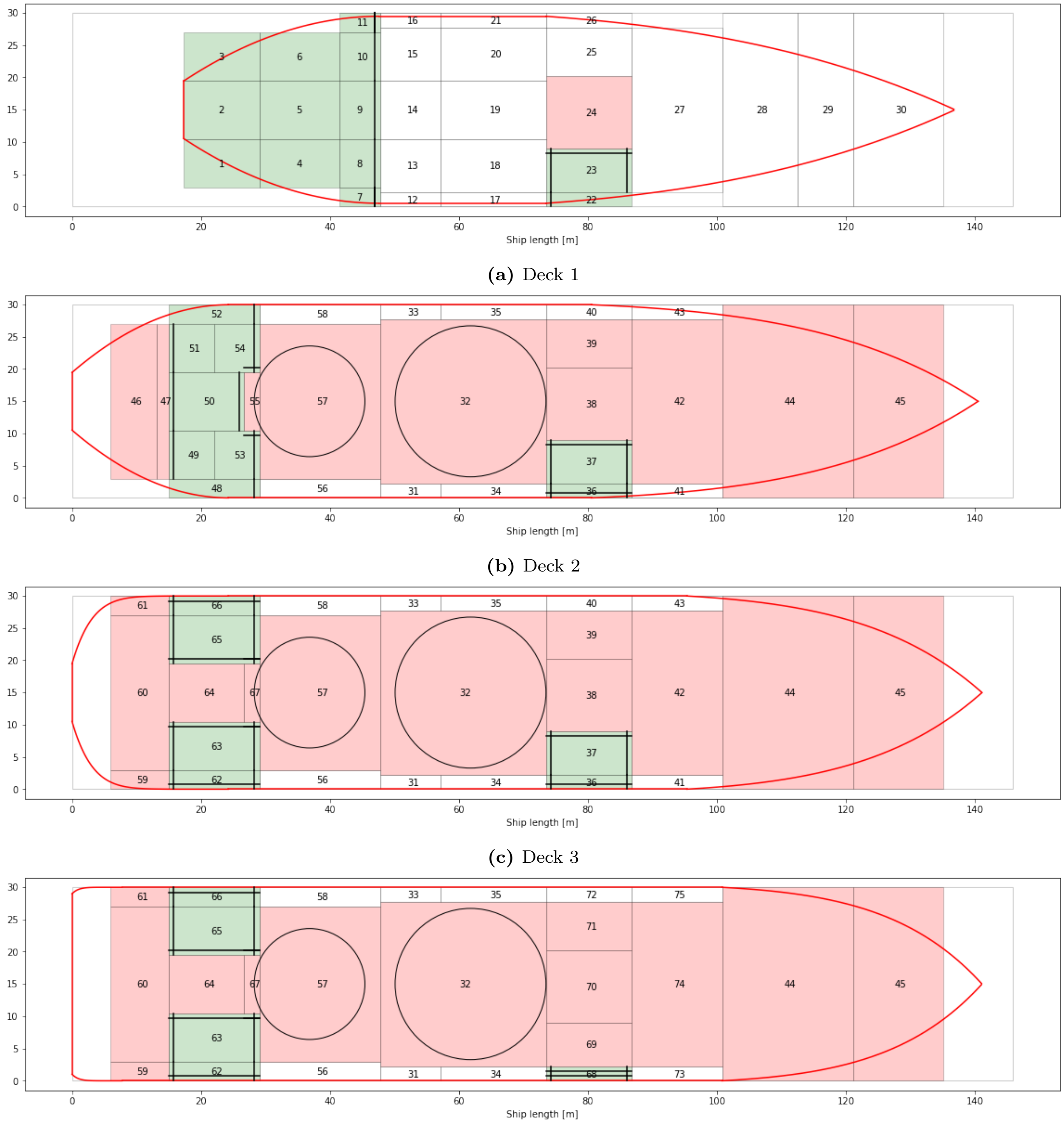
(a)

(b)

Figure 7: Figure 7a shows a distribution of the RO costs divided by the DO costs for all 2700 instances. Figure 7b shows boxplots that represent a comparison of all three models in the SO objective.

4.3 Realistic instance: an example of an application

In this section, we apply the three models to a realistic graph to show what robust pipe routing looks like in practice. We collaborated with a shipyard to study a ship consisting of four decks. This graph is based on a schematic overview of [29], as shown in Figure 8, which computes possible locations of the methanol tanks in a ship currently fueled by diesel. Note that the figures differ slightly from the original paper due to an improvement in the methodology. This ship contains a moonpool, which is an opening in the floor that gives access to the water below, enabling operators to lower tools into the sea. Pipes cannot go through these rooms, making pipe routing more difficult, as it restricts routes. A cargo ship hold would do the same, as pipes cannot enter the cargo space. However, a work ship like this gives more variation since cargo ships have limited spaces and equipment in front of the cargo hold. We represent the 3D network of compartments in the ship as a graph; vertices denote rooms, and edges represent connections between



(a) Deck 1

(b) Deck 2

(c) Deck 3

(d) Deck 4

Figure 8: Figures 8a-8d shows an overview of the four decks of the ship from [29], where each number represents a room. Green rooms denote methanol tanks, while red rooms cannot serve as methanol tanks as they are already occupied. Rooms 24, 38, and 70 denote the moonpool.

adjacent rooms. The resulting graph contains 75 vertices and 156 edges. We use the Manhattan distance between the vertices for the costs γ_{1uv} of installing a single-walled pipe in the first stage. The installation costs of double-walled pipes are again $\gamma_{2uv} = 2\gamma_{1uv}$, $\forall (u,v) \in \mathcal{E}$.

We assume that we start with diesel in the first stage and transition to diesel or methanol in the second stage. We use the locations of the current diesel tanks of the ship as terminals for the diesel scenario and the locations of the methanol tanks computed in [29] as terminals for the methanol scenario. Diesel pipes cannot be routed through the double bottom or through rooms adjacent to the water. Whereas diesel can be routed through either single- or double-walled pipes, methanol requires double-walled pipes due to safety regulations [28]. A mathematical overview of the realistic instance is shown below:

- $\mathfrak{T} = \mathfrak{T}^{(1)} = \{37, 42, 53, 54, 63, 65\}$
- $\mathfrak{T}^{(2)} = \{1 - 11, 22, 23, 36, 37, 42, 48 - 54, 62, 63, 65, 66, 68\}$
- $\mathfrak{P} = \mathcal{P} = \mathcal{P}^{(1)} = \{1, 2\}$
- $\mathcal{P}^{(2)} = \{2\}$
- $\mathcal{F} = \{1 - 30, 31, 33, 34, 35, 36, 40, 41, 43, 44 - 48, 52, 56, 58, 59, 61, 68, 72, 73, 75\}$
- $\mathfrak{E} = \mathfrak{E}^{(1)} = \{(u,v) : u \in \mathcal{V} \setminus \mathcal{F}, v \in \mathcal{V} \setminus \mathcal{F}, u \sim v, u < v\}$
- $\mathfrak{E}^{(2)} = \mathcal{E}$,

where vertex 42 represents the engine room and set $\mathcal{F}$ represents the rooms in which we cannot install diesel pipes. We ran all models on the realistic graph and show our findings in Table 6. For SO, we assume equal probabilities for the scenarios in the second stage, i.e., $\rho_1 = \rho_2 = \frac{1}{2}$. We see that the models with directed formulations require more compilation time, probably caused by an increase in the number of decision variables and constraints, but yield considerably shorter run times. When focusing on the undirected formulations, we see that the best integer solution is found relatively quickly and that the solver needs a relatively long time to close the gap, whereas directed formulated models close the gap within almost a second.

	Deterministic		Stochastic		Robust	
	(DO-U)	(DO-D)	(SO-U)	(SO-D)	(RO-U)	(RO-D)
Compilation time	0.034	0.054	0.361	0.593	0.359	0.566
Run time	0.127	0.045	450.333	0.894	566.527	1.137
Decision variables	1,092	1,405	10,920	12,171	10,921	12,172
Constraints	765	1,609	8,391	14,957	8,393	14,959

Table 6: Result overview of running all models on the realistic graph.

The optimal routes according to DO and RO are displayed in Figure 10. We did not include the optimal route according to SO as it is similar to RO's solution. All three figures use different vertex colors denoting different connections with lower and upper decks. In Figure 9, we see that the DO solution only contains single-walled pipes and avoids rooms adjacent to the water. Figure 10a and 10b show that the RO solution uses single and double-walled pipes to prepare for diesel and methanol. The RO solution shows three main insights: 1) the pipe route goes via the starboard side of the moonpool to the engine room due to room 37, which contains a tank as well; 2) the double bottom contains a center pipeline that connects rooms 1-11 with room 23, which is adjacent to room 37; and 3) single-walled pipes are placed to connect the diesel tanks to the engine room, as they cannot use the aforementioned pipeline in the double bottom.

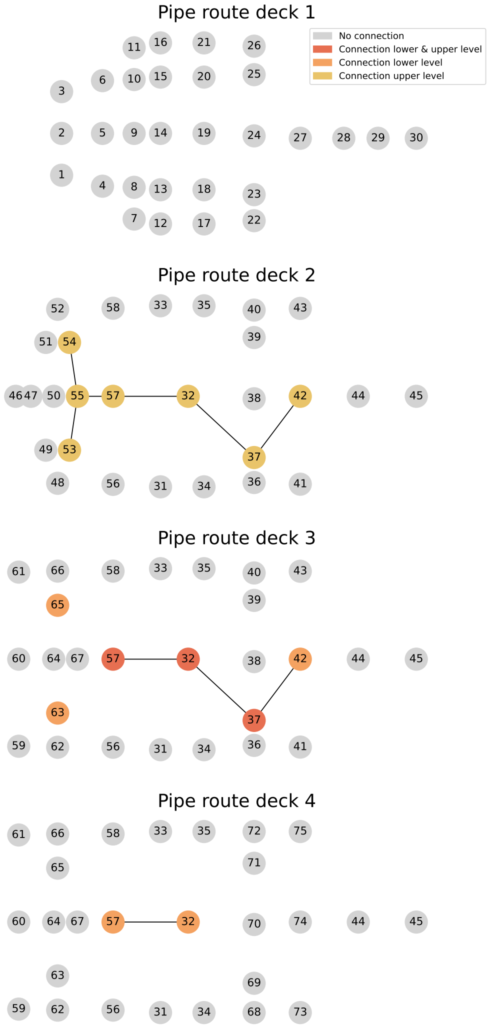
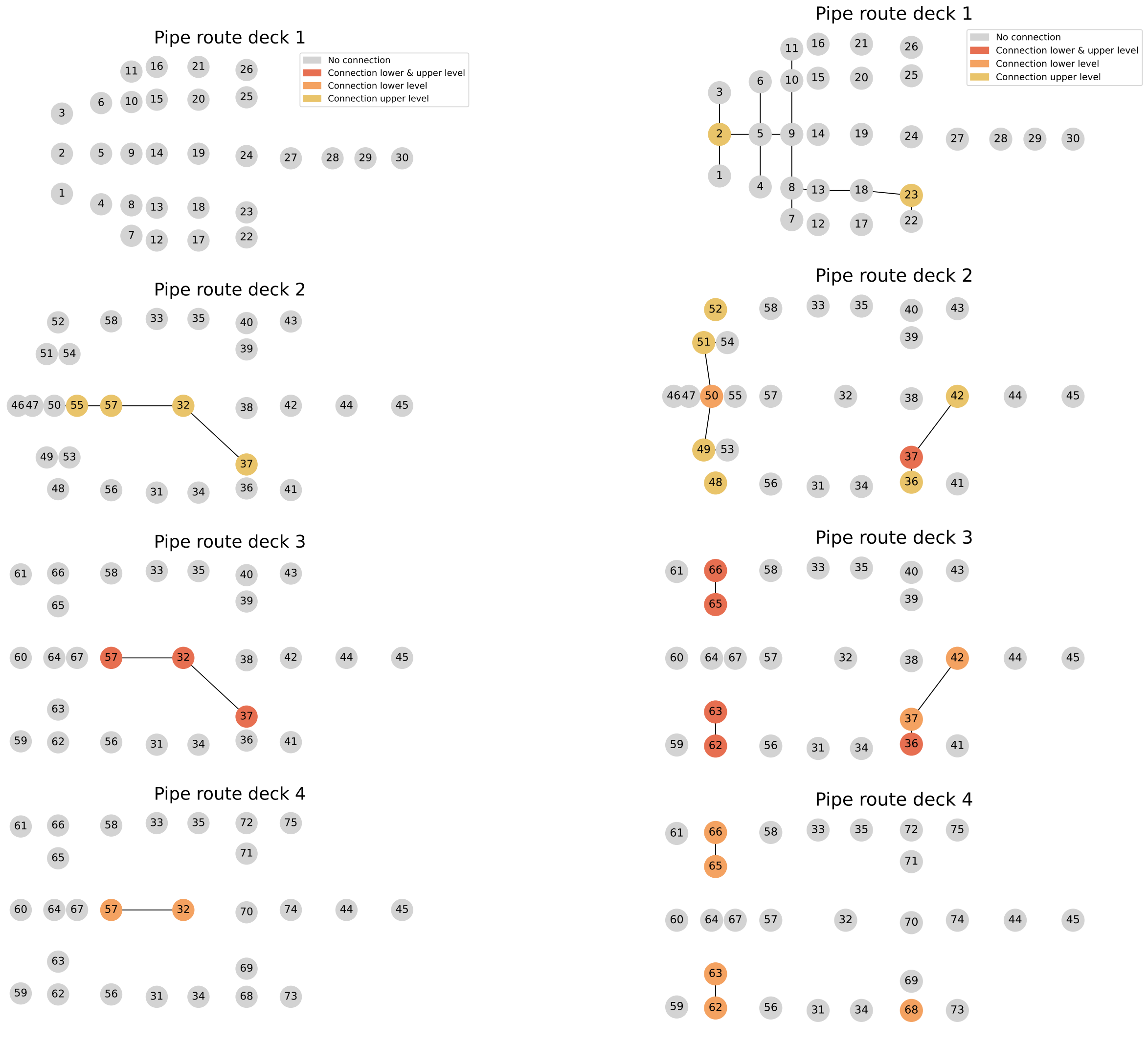


Figure 9: Optimal DO pipe route for the realistic graph, which only consists of single-walled pipes.



(a) **(b)**

Figure 10: Optimal RO pipe route for the realistic graph. Figure 10a shows the single-walled pipe route, whereas Figure 10b represents the double-walled pipe route.

5 Conclusion

Motivated by ship pipe routing under the uncertainty of the energy transition, we have introduced the 2S-SSFP and corresponding DO, SO, and RO models. For each of these three models, we have used both undirected and directed flow formulations based on [34]. We have applied them to three graphs to highlight different aspects of the 2S-SSFP and the proposed models.

Our experiments show that considering uncertainty can yield relative gains up to 22%. An application to a graph representing an actual ship shows three insights: 1) we install double-walled pipes in the center of the double-bottom to connect methanol tanks with the engine room; 2) we use a combination of single- and double-walled pipes on the other decks, and 3) we route the pipe network via the starboard side. Furthermore, we find that DO is typically the cheapest in the first stage but has considerable second-stage costs. SO and RO invest more in the first stage, which results in lower second-stage costs. DO performs the worst when the first stage contains terminals that lie close to each other, whereas the second stage consists of widespread terminals over the graph. The run times of the models increase when the number of scenarios, terminals, and terminal groups increases, especially when the terminals are widespread over the graph. More specifically, the directed formulations require longer compilation times but yield considerably shorter run times than the undirected formulations, which is in line with the findings in [34]. The difference in run times between directed and undirected formulations increases as the number of terminals increases.

For future research, we suggest studying methods that perform well on larger graphs to take more ship details into account, as an ILO model typically does not scale well. Possible scaleable methods are (meta-)heuristics or the L-shaped method for SO. Additionally, we could make the model more realistic by allowing multiple fuel types within one scenario. The energy transition might consist of multiple stages with different scenarios. Consequently, multi-stage SO could be helpful in this case. Finally, we could make the costs of installing pipes dependent on their location on the ship. For example, installing pipes near an engine room might be more expensive due to safety measures.

Declaration of interests

The authors declare that there are no known conflicts of interest that can influence the work presented in this research.

Acknowledgements

We thank Jesper Zwaginga for providing the data and helping with the first version of the mathematical models. We thank Joris Slootweg and Ruurd Buijs for their insightful input and feedback. Additionally, we thank SURF (`www.surf.nl`) for the support in using the National Supercomputer Snellius. This publication is part of the project READINESS with project number TWM.BL.019.002 of the research program *Topsector Water & Maritime: the Blue route* which is partly financed by the Dutch Research Council (NWO).

References

[1] A. Asmara. *Pipe routing framework for detailed ship design.* VSSD, Delft, 2013. OCLC: 864752777.

[2] M. Bateni, M. Hajiaghayi, and D. Marx. Approximation Schemes for Steiner Forest on Planar Graphs and Graphs of Bounded Treewidth. *Journal of the ACM*, 58(5):1–37, Oct. 2011.

[3] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization:.* Princeton University Press, Dec. 2009.

[4] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming.* Springer Series in Operations Research and Financial Engineering. Springer New York, New York, NY, 2011.

[5] M. Blokland, R. D. Van Der Mei, J. F. J. Pruyn, and J. Berkhout. Literature Survey on Automatic Pipe Routing. *Operations Research Forum*, 4(2):35, Apr. 2023.

[6] I. Bomze, M. Chimani, M. Jünger, I. Ljubić, P. Mutzel, and B. Zey. Solving Two-Stage Stochastic Steiner Tree Problems by Two-Stage Branch-and-Cut. In O. Cheong, K.-Y. Chwa, and K. Park, editors, *Algorithms and Computation*, volume 6506, pages 427–439. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. Series Title: Lecture Notes in Computer Science.

[7] O. Dinu and A. M. Ilie. Maritime vessel obsolescence, life cycle cost and design service life. *IOP Conference Series: Materials Science and Engineering*, 95:012067, Nov. 2015.

[8] M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness.* A series of books in the mathematical sciences. Freeman, New York [u.a], 27. print edition, 2009.

[9] E. Gassner. The Steiner Forest Problem revisited. *Journal of Discrete Algorithms*, 8(2):154–163, June 2010.

[10] L. Ghalami and D. Grosu. Approximation algorithms for Steiner forest: An experimental study. *Networks*, 79(2):164–188, Mar. 2022.

[11] B. L. Gorissen, I. Yanikoglu, and D. Den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, June 2015.

[12] M. Groß, A. Gupta, A. Kumar, J. Matuschke, D. R. Schmidt, M. Schmidt, and J. Verschae. A Local-Search Algorithm for Steiner Forest. 2017. Publisher: arXiv Version Number: 1.

[13] A. Gupta, M. Hajiaghayi, and A. Kumar. Stochastic Steiner Tree with Non-uniform Inflation. In M. Charikar, K. Jansen, O. Reingold, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 4627, pages 134–148. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. Series Title: Lecture Notes in Computer Science.

[14] A. Gupta and A. Kumar. A constant-factor approximation for stochastic Steiner forest. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 659–668, Bethesda MD USA, May 2009. ACM.

[15] A. Gupta and A. Kumar. Greedy Algorithms for Steiner Forest. In *Proceedings of the forty-seventh annual ACM symposium on Theory of Computing*, pages 871–878, Portland Oregon USA, June 2015. ACM.

[16] A. Gupta and M. Pál. Stochastic Steiner Trees Without a Root. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming*, volume 3580, pages 1051–1063. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. Series Title: Lecture Notes in Computer Science.

[17] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: approximation algorithms for stochastic optimization. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 417–426, Chicago IL USA, June 2004. ACM.

[18] A. Gupta, R. Ravi, and A. Sinha. LP Rounding Approximation Algorithms for Stochastic Network Design. *Mathematics of Operations Research*, 32(2):345–364, May 2007.

[19] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[20] P. Hokama, M. C. S. Felice, E. C. Bracht, and F. L. Usberti. A Heuristic Approach for the Stochastic Steiner Tree Problem. 2014.

[21] International Maritime Organization. Fourth IMO Greenhouse Gas Study, 2020.

[22] International Maritime Organization. IMO Strategy on reduction of GHG emissions from ships, July 2023.

[23] W. K. Klein Haneveld, M. H. v. d. Vlerk, and W. Romeijnders. *Stochastic programming: modeling decision problems under uncertainty.* Graduate texts in operations research. Springer, Cham, 2020.

[24] M. Leitner, I. Ljubić, M. Luipersbeck, and M. Sinnl. Decomposition methods for the two-stage stochastic Steiner tree problem. *Computational Optimization and Applications*, 69(3):713–752, Apr. 2018.

[25] E. Lindstad, B. Lagemann, A. Rialland, G. M. Gamlem, and A. Valland. Reduction of maritime GHG emissions and the potential role of E-fuels. *Transportation Research Part D: Transport and Environment*, 101:103075, Dec. 2021.

[26] I. Ljubić. Solving Steiner trees: Recent advances, challenges, and perspectives. *Networks*, 77(2):177–204, Mar. 2021.

[27] I. Ljubić, P. Mutzel, and B. Zey. Stochastic survivable network design problems: Theory and practice. *European Journal of Operational Research*, 256(2):333–348, Jan. 2017.

[28] Lloyd's Register. Rules and Regulations for the Classification of Ships, 2023.

[29] M. Minderhoud. A Real Options Approach to determine the Value of Design-for-Conversion under Uncertainty. Feb. 2023.

[30] J.-H. Park and R. L. Storch. Pipe-routing algorithm development: case study of a ship engine room design. *Expert Systems with Applications*, 23(3):299–309, Oct. 2002.

[31] M. Prussi, N. Scarlat, M. Acciaro, and V. Kosmas. Potential and limiting factors in the use of alternative fuels in the European maritime sector. *Journal of Cleaner Production*, 291:125849, Apr. 2021.

[32] J. Pruyn. Are the new fuel-efficient bulkers a threat to the old fleet? *Maritime Business Review*, 2(3):224–246, Sept. 2017.

[33] J. F. J. Pruyn. Benchmarking bulkers delivered between 2010 and 2016, identifying the effect of the EEDI introduction. *Journal of Shipping and Trade*, 5(1):11, Dec. 2020.

[34] D. Schmidt, B. Zey, and F. Margot. Stronger MIP formulations for the Steiner forest problem. *Mathematical Programming*, 186(1-2):373–407, Mar. 2021.

[35] G. Schäfer. Steiner Forest: 1995; Agrawal, Klein, Ravi. In M.-Y. Kao, editor, *Encyclopedia of Algorithms*, pages 897–900. Springer US, Boston, MA, 2008.

[36] C. Swamy and D. B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. *ACM SIGACT News*, 37(1):33–46, Mar. 2006.

[37] Xiao-long Qian, Tao Ren, and Cheng-en Wang. A survey of pipe routing design. In *2008 Chinese Control and Decision Conference*, pages 3994–3998, Yantai, Shandong, China, July 2008. IEEE.

[38] B. Zey. ILP formulations for the two-stage stochastic Steiner tree problem, Nov. 2016. arXiv:1611.04324 [cs].

[39] J. J. Zwaginga and J. F. J. Pruyn. An Evaluation of Suitable Methods to Deal with Deep Uncertainty Caused by the Energy Transition in Ship Design. In *Day 2 Mon, June 27, 2022*, page D021S003R002, Vancouver, Canada, June 2022. SNAME.

[40] A. Çivril. Approximation of Steiner forest via the bidirected cut relaxation. *Journal of Combinatorial Optimization*, 38(4):1196–1212, Nov. 2019.