Ferry Service Network Design for Kiel fjord

Ingvild Eide Aslaksen[1], Elisabeth Svanberg[1], Kjetil Fagerholt[1], Lennart Christian Johnsen[2](✉), and Frank Meisel[2]

[1] Department of Industrial Economics and Technology, Norwegian University of Science and Technology, Trondheim, Norway
kjetil.fagerholt@ntnu.no

[2] Faculty of Business, Economics and Social Science, Christian-Albrechts-University Kiel, Kiel, Germany
{lennart.johnsen,meisel}@bwl.uni-kiel.de

Abstract. This paper considers a ferry service network design problem using autonomous ferries for the practical case of the Kiel fjord. Among others, the city of Kiel, Germany, currently runs a number of initiatives for developing an autonomous ferry system to open up new mobility opportunities. The city is divided by the Kiel fjord into an eastern and a western part and the current infrastructure is mainly built to accommodate car transportation on roads around the fjord. We provide a new optimization model for the generation of schedules for an autonomous ferry service, including route design and determination of departure frequencies. The model captures practically relevant aspects of minimum required departure frequencies between specific port pairs and understandable ferry schedules, whilst maximizing customer service quality (i.e., excess transit times and departure frequencies). We provide a two-step optimization approach where candidate combinations of routes and departure frequencies are heuristically generated a priori and fed into an integer programming model. Experiments on real world data provide managerial insights in regard to ferry fleet size, port network design and ferry schedules.

Keywords: Network design problem · Autonomous ships · Kiel fjord

1 Introduction

Many countries currently develop autonomous ferry systems to open up new mobility opportunities in coastal areas. A central idea is to replace the existing conventional ferries (which are often large units to achieve economies of size) by smaller autonomous units that can be deployed more flexibly. Among others, the city of Kiel, Germany, currently runs a number of initiatives for developing

L.C. Johnsen—Supported by DFG - Deutsche Forschungsgemeinschaft through project GZ: ME 3586/2-1.

E. Lalla-Ruiz et al. (Eds.): ICCL 2020, LNCS 12433, pp. 36–51, 2020.
https://doi.org/10.1007/978-3-030-59747-4_3

such an autonomous ferry system. The city is divided by the Kiel fjord into an eastern and a western part. Up to now, two scheduled ferry lines bring people from east to west and vice versa, see Fig. 1. Each line is operated by one to three large ships with a capacity of 200 passengers each. The service frequency at the mooring spots is relatively low (one ship every 15 to 120 min) which is little attractive for customers.

Autonomy means that operations happen automatically, controlled by machines, and not humans [5]. Hence, a fully autonomous ferry could be operating on water without any captain or other crew stationed at the ferry. This facilitates new cost structures, and can enable the use of several smaller ferries, thus providing a more flexible and rapid ferry service offering. The technology, documentation and regulations needed for autonomous transportation are yet to some extent undeveloped [6]. However, the interest for the technology is high, and in 2018, Rolls-Royce and Finferries conducted a demonstration of the world's first fully autonomous ferry with 80 passengers on board. Mikael Makinen, Rolls-Royce President Commercial Marine, claims that “the demonstration proves that the autonomous ship is not just a concept, but something that will transform shipping as we know it” [13].

This paper presents and discusses solution methods regarding the design of routes for an autonomous ferry service, including the selection of departure frequencies, for a practical case at the Kiel fjord. This problem is henceforth

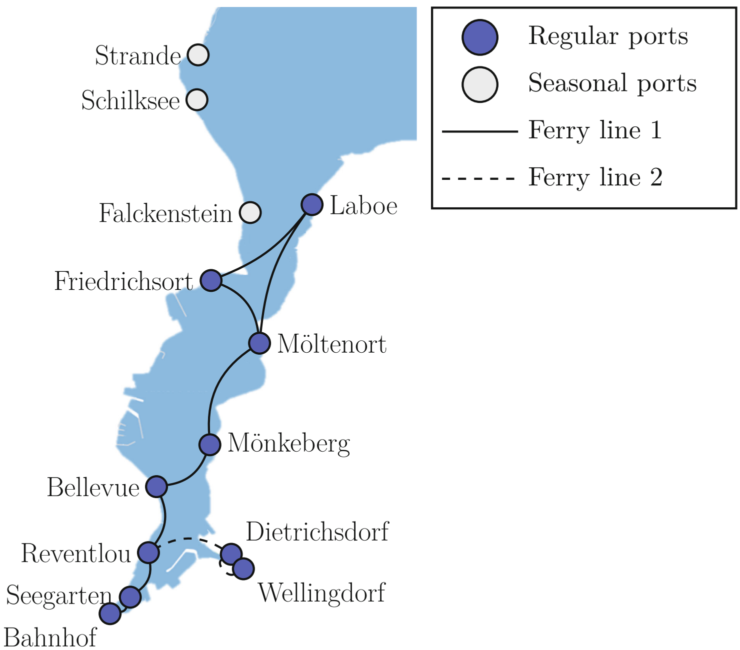


Fig. 1. Current ferry service at the Kiel fjord

referred to as the Ferry Service Network Design Problem (FSNDP). The paper contributes to a larger project at Kiel University, named *CAPTin Kiel*, "Clean Autonomous Public Transport in Kiel". The aim is to provide decision support to the implementation of a ferry service network for passenger transportation in the Kiel fjord using autonomous passenger ferries.

This paper makes the following contributions. We propose a new integer programming (IP) model, which can provide decision support for the FSNDP. The problem is solved in a two-step optimization approach where candidate combinations of routes and departure frequencies are generated a priori and fed as input to the IP model. To solve realistic problem instances, we provide a heuristic procedures to effectively generate candidate combinations of routes and departure frequencies. A computational study is provided using practical test instances based on the Kiel fjord in order to evaluate the optimization approach and to provide managerial implications.

The outline of this paper is as follows. Section 2 presents a review of relevant literature. A formal description of the FSNDP is presented in Sect. 3 including the IP model. Section 4 describes how the combinations of routes and departure frequencies are generated. A computational study is presented in Sect. 5. Lastly, Sect. 6 summarizes the managerial insights and concludes the paper.

2 Related Literature

[9] were the first ones to formulate a ferry service network design problem (FSNDP). They formulated a tactical problem to optimize the fleet size, routing and scheduling of a ferry service. They also formulated a multi-objective optimization model where they seek to minimize operator cost in terms of the fleet size, trip operating cost and negative revenue, and user cost in terms of waiting time and a penalty for multi-stops. [15] present an extension of the work of [9] by introducing a heterogeneous fleet as well as heterogeneous customer preferences. [10] introduce stochastic demand to the FSNDP, and formulate a two-stage stochastic model where they first determine routes to cover a given percent of the expected demand, and then, when the actual demand is revealed, offer an ad-hoc service to cover the remaining demand. [1] wrote an extension on this, by adding user equilibrium. [12] formulate a robust modelling of the service network design problem, and conducts a case study based on cases presented in [9] and [15]. They assume that only an upper bound and the mean of the passenger demand is known. The case study showed that using "loose information" in the absence of more exact values could lead to higher cost, which motivates more effort in obtaining accurate demand data when designing passenger transit routes. The most recent literature on the FSNDP is provided by [3]. They present a method to find the maximum passenger utility spanning tree which connects all ports. The decisions in the problem are which pairs of ferry stations should be directly connected and where the ferry hubs should be located, and the objective is to maximize passenger utility (minimize some function of transit time). They use the entropy maximization (EM) method to create a logit choice model

which in turn generates a random utility interpretation, and can then optimize an expected passenger utility. The problem is a strategic network design, hence disregarding frequencies and ferry capacity. They present two greedy heuristic approaches to solve the problem with up to 36 ports.

Network design problems are in general NP-hard [2]. Heuristics seem to be the general trending solution method. [11] propose three MIP-based heuristics: local branching, variable neighborhood branching, and variable neighborhood decomposition search. They find that variable neighborhood branching outperforms the other two heuristics as well as the commercial Cplex MIP solver. [8] propose a variable neighborhood search heuristic for the ship routing and scheduling problem with split loads. The heuristic provides good solutions and solves real-life instances within reasonable time. In our paper, we aim to find the sweet spot between tactical and strategic planning; taking it a step further than [3] by identifying specific routes and frequencies, but still disregard ferry capacity and not model exact load, thus keeping the model simple enough to be solved exact. The FSNDP is solved in a two-stage optimization procedure in the manner that a set of candidate routes are first generated, and then an optimal subset of these routes are chosen simultaneously with their respective frequencies. Therefore, lastly, we briefly discuss some literature on route generation. [7] present a heuristic which merges the two steps, thus the output being a final set of routes with associated frequencies. They start with an initial set of routes and generate several routes that can replace routes in the initial set if they meet some criteria and yield better solutions. For each pair of origin and destination, they generate two candidate routes; the shortest path (note that not all nodes are directly connected) and the shortest path which is sufficiently different from the first one. This ensures that the nodes with most demand get the most direct routes, whilst keeping the routes in the network sufficiently different, thus ensuring more connectivity in the network and good connections also between nodes with less demand. To the best of our knowledge within ferry service literature, no one has created a route generation algorithm with a rule-based heuristic which aims to reflect the geography of the case study. Moreover, we provide a new simple model formulation that represents the level of customer service through the trade-off between departure frequencies and excess transit time, while ensuring a satisfactory amount of departures between important ports.

3 Modelling of the FSNDP

Based on the categorization of algorithms for solving network design problems introduced by [4], the proposed model is a type of "selection of routes". Feasible routes are generated a priori, and the FSNDP seeks to find the best combination of routes and frequencies to transport the passengers through the network. The FSNDP considers a homogeneous fleet of autonomous ferries, where the size of the fleet is the only attribute of interest. Ferries repeat an assigned route, which is a cyclic sequence of port visits, i.e., they start and end in the same port. These routes are continuously repeated, and the ferries always travel their assigned route. Passenger demand exists between each pair of ports in the network, and the FSNDP aims to offer a service without the use of transfers. Assuming passengers can board only a single ferry, excludes the possibility of a hub structure. To avoid excessively long detours, a maximum transit time is imposed on the definition of *serving* a port pair, and it compares the actual transit time with the shortest possible transit times, i.e. the direct connection.

The FSNDP has a multi-objective approach which represents the trade-off between rapid departure frequencies and short transit times. It seeks to maximize perceived customer service, and in particular *departure frequency*, i.e., how often the ferries depart, and *transit times*, i.e., the time it takes for a passenger to travel in the ferry network. Customer service is represented by some artificial *user utility* for each combination of a route (r) and a frequency (f), which we will denote an rf-combination. In addition, each pair of origin and destination ports (OD-pair) is associated with a *minimum required frequency* ensuring certain service levels in the network. These minimum frequencies guarantee that a ferry travels between these ports at least a certain number of times per hour, regardless of which route it is assigned to. Moreover, to secure that the ferry schedule is understandable for the passengers, a maximum allowed number of unique routes included in the ferry schedule is imposed.

The route network is designed holistically, hence without modeling ferry specific passenger flow. The triangle inequality is satisfied for all transit times. This implies that an indirect route always yields higher transit times than a direct between the same pair of ports. Every route has a round-trip time, i.e., the time required for the ferry to traverse the route once. In order to satisfy the chosen departure frequency, a route may incur a waiting time after finishing the route. The model generates a network design for only one hour. However, the results can be used for the entire time horizon by repeating the solution. Therefore, we also assume that the input parameters are deterministic and constant within the considered time horizon. Moreover, we assume constant transit times, i.e the transit times are not influenced by e.g. weather conditions. The mathematical model with its notation summarized in Table 1 is as follows.

Objective

$$\max \text{z} = \sum_{r \in \mathcal{R}} \sum_{f \in \mathcal{F}_r} \sum_{i \in \mathcal{P}_r} \sum_{j \in \mathcal{P}_r} A_{rfij} (U_f^F + U_{rfij}^T) D_{ij} \, x_{rf} \tag{1}$$

Table 1. Summary of notation

Indices:	
r	Route
i, j	Port
f	Departure frequency, per hour
Sets:	
$\mathcal{P}$	Set of ports
$\mathcal{P}_r$	Set of ports in route r
$\mathcal{R}$	Set of routes
$\mathcal{F}_r$	Set of available departure frequencies for route r
Parameters:	
V	Number of ferries available
A_{rfij}	Number of times route r with frequency f serves the port pair (i, j)
F_{ij}	Minimum frequency of departures from port i to j, per hour
D_{ij}	Demand from port i to port j per hour
R^{Max}	Maximum number of unique routes allowed
T_r	Total transit time of completing a round trip of route r
T_{rf}^{Wait}	Waiting time for a ferry on route r with departure frequency f
U_f^F	Utility associated with departure frequency f
U_{rfij}^T	Utility associated with excess transit time for port pair (i, j) in route r with frequency f
Decision variables:	
x_{rf}	1 if route r is served with frequency f, 0 otherwise

Constraints

$$\sum_{r \in \mathcal{R}} \sum_{f \in \mathcal{F}_r} A_{rfij} \, f \, x_{rf} \geq F_{ij}, \qquad (i, j) \in \mathcal{P}, \, i \neq j \tag{2}$$

$$\sum_{f \in \mathcal{F}_r} x_{rf} \leq 1, \qquad r \in \mathcal{R} \tag{3}$$

$$\sum_{r \in \mathcal{R}} \sum_{f \in \mathcal{F}_r} (T_r + T_{rf}^{Wait}) f \, x_{rf} \leq V \tag{4}$$

$$\sum_{r \in \mathcal{R}} \sum_{f \in \mathcal{F}_r} x_{rf} \leq R^{Max} \tag{5}$$

$$x_{rf} \in \{0, 1\} \qquad r \in \mathcal{R}, f \in \mathcal{F}_r \tag{6}$$

The objective function (1) selects the combination of routes and frequencies that maximizes user utility in the ferry network. Thus total user utility is modelled as a weighted sum of utility with respect to departure frequency, U_f^F, and to excess transit time, U_{rfij}^T. Let D_{ij} be an expected demand between port i and j and A_{rfij} be a parameter that states the number of times route r with frequency f serves port pair (i, j). The combination of D_{ij} and A_{rfij} is used to weigh U_f^F and U_{rfij}^T such that sought-after (i, j)-connections have more impact in the objective. Constraints (2) ensure every port-pair (i,j) is visited at least as often as the minimum required frequency F_{ij}. Constraints (3) ensure only one departure frequency is chosen per route r. Moreover, Constraint (4) ensures that the total number of required ferries does not exceed the number of ferries available. The transit time T_r of the route r is calculated by summing up the direct transit times and the berth times along the route r. The waiting time is calculated by $T_{rf}^{Wait} = \frac{\lceil f\, T_r \rceil - f\, T_r}{f}$. As an example, consider a transit time of 20 min (1/3 h) and a frequency of 4, the waiting time is $\frac{\lceil 4 \cdot \frac{1}{3} \rceil - 4 \cdot \frac{1}{3}}{4} = \frac{\frac{2}{3}}{4} = \frac{1}{6} \implies 10\,\text{min}$. Then, with transit and waiting times given in hours and frequency f given in hours^{-1}, the product of times and frequency yields the number of ships to deploy on a route. Constraint (5) limits the number of unique routes in the network. Lastly, Constraints (6) define the feasible area for the decision variables.

4 Generation of Route and Frequency Combinations

In this chapter, we first propose rules for which to generate the routes and, afterwards, we present an algorithm to generate candidate combinations of routes and departure frequencies, denoted rf-combinations.

Our route generation procedure deploys a *chain* structure, which implies that each port can be visited at most twice in a route [14]. One way to generate the candidate routes is to combine the ports in all possible sequences for all route lengths, but the generation would experience a combinatorial explosion. We seek to generate as few routes as possible, while ensuring good candidate routes remain included. Therefore, we develop a route generation heuristic which aims to identify candidate routes that are deemed reasonable with respect to passenger transportation. The heuristic evaluates each route independently, and it is constructed such that generated candidate routes satisfy the following rules:

Rule 1 Do not generate identical route cycles.
Rule 2 Include only one directional direct link per port pair in the route.
Rule 3 Routes must contain at least one "large" port.
Rule 4 "Adjoining pairs" must be visited consecutively.
Rule 5 Disallow north/south "zigzagging" in the routes.

The first rule implies that a route with the same ports and visiting sequence, but with a different starting port, will not be included. As an example, consider the three routes $[A, B, C, A]$, $[B, C, A, B]$ and $[C, A, B, C]$. These routes are similar except for that they have different starting ports. The starting port is here defined as the port along the route where all the waiting time is allocated to. So by determining which port to have as starting port, so as to maximize the utility of the route, we can also determine which of these three routes to include as a candidate route. If for example port A is chosen as the starting port, only route $[A, B, C, A]$ will be included.

The second rule restricts the number of direct connections between a port pair in a route, thus disallowing e.g. $[A, B, C, A, B]$ since passengers going between ports A and B may as well enter the ferry on the first direct link.

The third rule aims to eliminate routes with low demand. We denote ports located in busy areas as *large ports*, i.e., ports with high demand, and rule three states that all routes must contain at least one of those large ports.

The fourth rule aims to avoid detours for ports located close to each other, i.e., *adjoining port pairs*. The algorithm rejects a route if it contains both of the ports in the *adjoining port pair* but these are not visited right after each other.

Lastly, the fifth rule disallows "zigzagging" in the north/south direction of the fjord. Since the routes are cyclic and not necessarily start in the most northern or southern ports, we allow them to turn two times, e.g. the route first goes south, then north and then south again. If the route turns more than two times, it is discarded. The rule is based on the ports having an attribute related to level in the north/south direction, similar to latitude. Note that, the concept of zigzagging is allowed in the east/west direction, because the benefits of utilising a ferry instead of alternative land transportation is larger when traveling across the fjord rather than along it.

The candidate combinations of routes and departure frequencies are generated in a two-step procedure: In the first step, candidate routes are generated with feasibility checks according to rules 1 to 5. In the second step, combinations of those candidate routes with departure frequencies are proposed with feasibility checks according to waiting times. Algorithm 1 presents the pseudocode of the candidate route generation procedure. The procedure constructs routes attractively by extending the length of the route by one port per iteration, checking the feasibility of the extended route according to rules 1 to 5, and if it is feasible, the extended route is added to the set of candidate routes. In particular, the first iteration generates routes with two ports. Then, a third port is added to each of these routes, checked for feasibility according to the rules mentioned above, and if the new route is deemed feasible, it is saved as a candidate route. By only extending routes which are in themselves feasible, we avoid enumerating all permutations of the routes, thereby decreasing computational time required to generate the routes. One exception to the feasibility requirement concerns routes with equal last and second last visit, e.g. $[A, B, A, A]$. These have to be temporarily feasible to construct other routes, e.g. $[A, B, A, C, A]$, but they will be discarded after they have been extended, such that they are not considered candidate routes.

We recall that the different combinations of routes and frequencies yield different waiting times, as described in Section 3. Some of these combinations may be undesirable due to excessively long waiting times in the port between scheduled departures. Therefore, we only generate the combinations with waiting

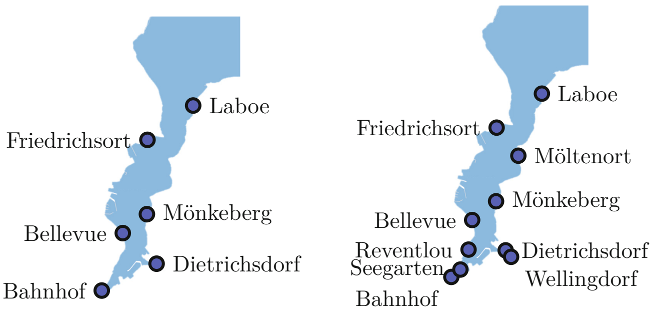
Algorithm 1: Candidate routes generation.

Initialize set of construction ports as all ports in the port case;
for *all ports in the port case* **do**
  if *the port is large* **then**
    create initial set of candidate routes by combining this port with each of the ports in the set of construction ports, except itself;
    for *routes in the set of candidate routes* **do**
      extend the route with each port in the construction port set;
      if *the extended route is feasible by all rules* **then**
        save the route in the set of candidate routes;
      else
        go to next route;
      end
    end
    remove this port from the set of ports to construct routes from;
  end
end

times less than a threshold, defined by W^{Max}, thus reducing the number of variables in the problem.

5 Computational Study

Our test instances are based on a practical case of the Kiel fjord. Kiel is a northern German city and houses nearly 250,000 people. Kiel is a seaport city, where the western and eastern part of the city are divided by the fjord. A map of the Kiel fjord with the ports is shown in Fig. 2b).



(a) Ports included in the *reduced* port case. **(b)** Ports included in the *full* port case.

Fig. 2. Illustration of the two port cases.

Table 2 lists all ports and their relevant attributes. Each port is defined as either "small" (S), "medium" (M), or "large" (L), and some ports are classified as "well connected" (*Well con.*), which are port pairs with excellent alternative on land connections, e.g., ports in walking distance. Both attributes will be reflected in demand generation. Adjoining ports (*Adj.*) are located directly next to each other. Further, ports have a defined location in the north/south (*N/S*) direction, which is related to their position on the map.

Table 2. Description of ports.

No.	Port name	Size	Well con.	Adj.	N/S
0	Laboe	L	-	-	3
1	Möltenort	M	-	-	5
2	Mönkeberg	M	-	-	6
3	Dietrichsdorf	L	4	4	8
4	Wellingdorf	L	3	3	8
5	Bahnhof	L	6, 7	6	10
6	Seegarten	S	5, 7, 8	5	9
7	Reventlou	L	5, 6, 8	8	8
8	Bellevue	S	6, 7	7	7
9	Friedrichsort	M	-	-	4

5.1 Instances

We present two different port cases, *reduced* and *full*. The *full* port case comprises all the ports used in today's ferry system, yielding a case with ten ports. The *reduced* case aims to create an alternative that covers the whole fjord, but with fewer ports, and thus it comprises six of the ten ports spread evenly across the fjord. The two port scenarios are visualized in Fig. 2a) and 2b). For each of these port cases, we define a set of test instances differing in the minimum required departure frequencies, the fleet size, and the maximum number of unique routes. Table 3 provides a summary of the test instances.

We construct three different settings for the minimum required frequencies per OD-pair. The minimum required frequency per OD-pair depends on the port size attribute. However, we choose to let the frequencies only depend on the size of the destination port. The minimum frequencies in the *moderate* setting are 1, 2, and 3 for *small/medium/large* ports, respectively. For the *relaxed* and *strict* settings we choose 0/1/2, and 2/3/4. For all *well connected* port pairs, the required frequency is set to zero, meaning we do not impose any requirement of visits between ports that are located close to each other. However, note that there may still exist demand between well connected ports, so the solution may

Table 3. Overview test instances.

Instance	Port case	Minimum frequency	Fleet size	Unique routes
Full port case				
Full-Mod-30-6	Full	Moderate	30	6
Full-Mod-20-6	Full	Moderate	20	6
Full-Mod-40-6	Full	Moderate	40	6
Full-Mod-30-3	Full	Moderate	30	3
Full-Mod-30-9	Full	Moderate	30	9
Full-Rel-30-6	Full	Relaxed	30	6
Full-Str-30-6	Full	Strict	30	6
Reduced port case				
Red-Mod-20-6	Reduced	Moderate	20	6
Red-Mod-30-6	Reduced	Moderate	30	6
Red-Mod-10-6	Reduced	Moderate	10	6
Red-Mod-20-3	Reduced	Moderate	20	3
Red-Mod-20-9	Reduced	Moderate	20	9
Red-Rel-20-6	Reduced	Relaxed	20	6
Red-Str-20-6	Reduced	Strict	20	6

offer a departure between the ports. Today, the ferries offer a departure frequency of less than once an hour for all OD-pairs, which relates to the challenges of the current ferry service. Therefore, all of our cases, except small ports in the *relaxed* case, are at least as good as the current offering, which is valuable when aiming to design a better ferry service.

Further, we solve the problem for a various number of available ferries, i.e., different values of V. We consider four settings with 10, 20, 30, and 40 ferries. Lastly, we solve the model for different values of maximum allowed number of unique routes in the solution, i.e., R^{Max}. We define settings with three, six and nine unique routes. For the instances of *reduced* port case, we run five samples of demand, whilst for *full* port instances three samples of demand are considered.

5.2 Parameters

To calculate direct transit times T_{ij}^{Direct} between all port pairs (i, j), *Google Maps* was used to find the distances between ports. The distances were not always a straight line, but the shortest distance that yielded a reasonable path given the restrictions of the fjord. We assume a sailing speed of 17 km/h for all instances, which is derived from the time table of the current ferry service. The transit times are then calculated by dividing the distances by speed. Based on empirical measurements conducted during a field trip, the berth time at each port is set to three minutes.

We assume that a passenger will not accept excess transit times of more than 100% of the direct travel time. This requirement must be fulfilled for a combination of route and frequency to satisfy demand between two ports, as given by the parameter A_{rfij}. The departure frequencies deemed feasible are restricted, because the ferry service aims to provide understandable schedules. For example, a frequency of seven times per hour would yield a route that departs every 8.57 min, which is not very intuitive. However, a frequency of six times per hour, would depart every tenth minute. Thus, the available departure frequencies (*departures/hour*) for the routes are 1, 2, 3, 4, 5, 6, and 8 implying that a ferry visits the ports every 60, 30, 20, 15, 12, 10 and 7.5 min.

Moreover, demand between each OD-pair is randomly drawn from uniform distribution in the interval [50, 100]. The demands are adjusted to account for differences in port size. Each sampled OD-pair demand is multiplied with the product of two factors. The first factor relates to the origin port, and it is set to 0.75, 1.00, and 1.25 for small, medium and large ports, respectively. The second factor depends on the destination port, and is set to 0.5, 1.0, and 1.5 for small, medium, and large ports, respectively. Furthermore, independent of port size, the demand between *well connected* ports is decreased by 90% to account for the low demand between ports located close to each other.

The users' utility, defined by U_f^F and U_{rfij}^T, is calculated as follows. The aim of our approach is to represent the trade-off between decreased transit time and increased departure frequency, weighted by demand on the different connections. The utility functions are based on s-curve shapes where the turning point can be interpreted as users' reference points.

For departure frequency utility U_f^F, we assume a turning point at a value of four. This implies that the marginal utility from higher departure frequencies diminishes after four departures per hour. Furthermore, the s-shape implies that two different routes with frequency of four times per hour, provide more utility than a single route with a frequency of eight times per hour. Thus, it aims to construct a network with more diverse connections between ports. We formalized the frequency utility as

$$U_f^F = \begin{cases} f \cdot (0.05\, f + 0.8), & \text{if } 0 \leq f \leq 4 \\ f \cdot (-0.05\, f + 1.2), & \text{if } 4 < f \leq 8 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where parameter values of the quadratic equations are defined such that the curves cross at the turning point, while ensuring that the slope of the curve is lower for departure frequency below four than above four.

U_{rfij}^T is the utility of excess transit time from port i to port j in route r with frequency f. It rewards decreases in transit time, i.e. more direct routes. We assume that passengers have a reference point for excess transit time at 40%. This preferences is justified as passengers know they make use of a public offering, implying that they to some extent are willing to accept a detour on their journey. The s-curve shape implies that decreasing user's excess ride time from 50% to 40% produces more utility gains than reducing it from 20% to 10%.

This is formalized in Eqs. (8), (9), and (10). U^T_{rfij} is a mean of U^T_{rfimjn}, where m and n are indices for *port calls* in the route for port i and port j respectively. For example, if port i is visited two times in route r, $m = 2$ indicates the second visit in port i in route r. M_i is the number of port calls, i.e. the number of visits of port i in route r, and N_j is the number of port calls in port j. T_{rfimjn} is the transit time when traveling from port i at visit m to port j at visit n in route r with frequency f. T^{RE}_{rfimjn} is the percentage of excess transit time, and it is computed in Eq. (10).

$$U^T_{rfij} = \frac{\sum_{m=1}^{M_i} \sum_{n=1}^{N_j} U^T_{rfimjn}}{M_i N_j} \tag{8}$$

$$U^T_{rfimjn} = \frac{1}{e^{\cdot T^{RE}_{rfimjn} - 0.4} + 1} \tag{9}$$

$$T^{RE}_{rfimjn} = \frac{T_{rfimjn} - T^{Direct}_{ij}}{T^{Direct}_{ij}} \tag{10}$$

5.3 Computational Results

In this section, we present and analyze the computational results. We use Gurobi Optimizer version 9.0.1 to solve the IP model. The procedure for the generation of route and frequency combinations has been implemented in Python 3.7. We performed all computations on a computer with a 2.7 GHz Intel Core i5 processor and 8 GB RAM which runs the Mac OS X El Capitan (version 10.11.6) operating system.

Table 4 displays the computational results of all test instances. All values reported are averages of the replications with different demand samples. The first column describes the test instances. The next column displays the objective value, z, while the two next show the run time of pre-processing, *Pre-proc*, and the run time for solving of the optimization problem with Gurobi, *cpu*, in minutes, respectively. Columns 5 to 8 present attributes of the solutions. Column 5, *Frequency*, shows the average departure frequency per hour across all port pairs in the network. Column 6, *Excess t.t.*, shows the average excess transit time for the *best* possible connection between the port pairs weighted by the demand between the port pairs. Column 7, *Unique routes*, gives the number of unique routes in the solution and the last column, *Call per port*, displays the average number of visits per port per hour.

Both when running the test instances for the *reduced* and the *full* port case, all instances were solved to optimality. The number of candidate routes for the *reduced* port case is 473, and the number of feasible *rf*-combinations is 2,177. For the *full* port scenario, the number of routes is 257,400 and the number of feasible *rf*-combinations is 1,155,059. We see from the table that the computational times are very small for the *reduced* port case. For the *full* port instances, the total run times are close to three hours. Moreover, within both port cases, the objective

Table 4. Results of all test instances.

Instance	z	Pre-proc, *min*	cpu, *min*	Frequency	Excess t.t, *min*	Unique routes	Calls per port
Full port cases							
Full-Mod-30-6	72,522	150.3	13.1	7.4	7.6	6	17.3
Full-Mod-20-6	47,651	150.3	26.6	4.7	7.6	6	11.6
Full-Mod-40-6	95,835	150.3	24.6	10.1	7.7	6	22.6
Full-Mod-30-3	67,289	150.3	9.5	7.3	7.8	3	16.8
Full-Mod-30-9	74,339	150.3	10.6	7.2	6.5	9	16.8
Full-Rel-30-6	73,422	150.3	14.1	7.6	7.6	6	17
Full-Str-30-6	71,014	150.3	14.9	7.3	7.7	6	16.8
Reduced port cases							
Red-Mod-20-6	31,261	0.09	0.004	7.9	3.6	5	13
Red-Mod-10-6	15,396	0.09	0.006	4	5.9	3	6.8
Red-Mod-30-6	45,346	0.09	0.003	11.9	4.3	6	20
Red-Mod-20-3	30,048	0.09	0.005	7.9	5.3	3	13.2
Red-Mod-20-9	32,122	0.09	0.005	7.3	2.8	8.2	12.5
Red-Rel-20-6	31,261	0.09	0.003	7.9	3.6	5	13
Red-Str-20-6	31,137	0.09	0.006	7.8	3.8	5	13

value seems to be mostly affected by the number of ferries available. This is natural as both short transit times and frequent departures are dependent of the number of ferries.

We next investigate managerial implications. The average departure frequency between every port pair (weighted by demand) seems to be mostly affected by the number of ferries available. With 20 ferries (Full-Mod-20-6) it lies at 4.7 times per hour for the *full* port case. Increasing the number of ferries to 30 increases the departure frequency to about 7.4 and with even 10 more ferries, we get around 11.9. The results for the *reduced* port case resemble these effects. Interestingly, we see that the departure frequency is also strongly affected by the number of ports as we observe that the departure frequency increases from 4.7 to 7.9 when moving from instance Full-Mod-20-6 to Red-Mod-20-6. It seems that in the *full* port cases about 10 ferries more are required to reach about the same departure frequencies as in the *reduced* port cases.

Moreover, the excess transit times in the *full* port case instances are especially good for the case when we allow a large number of unique routes (Full-Mod-30-9). This implies that allowing many different routes in the schedule offers at least one direct connection for more port pairs. Comparing the port cases (Full-Mod-20-6 with Red-Mod-20-6 and Full-Mod-30-6 with Red-Mod-30-6), we see that the average excess transit times are somewhat higher in the *full* port cases. This makes sense as there are more ports that have to be visited in the routes in order to cover the minimum required frequencies. This indicates that there are fewer direct connections in the *full* port cases than in the *reduced* cases. However, from comparing Full-Mod-30-6 with Full-Mod-30-9, we observe that also in the *full* port cases many unique routes seem to have a positive effect.

6 Conclusion

Our computational study provides several managerial insights for increasing the perceived customer service (i.e., excess transit times and departure frequencies) in an autonomous ferry system. First, we observe that with fewer, more relevant and less allied ports, the great demand from these ports will be served more efficiently. While fewer ports means longer distances from passengers' origins and destinations to the ports, this gap can be easily closed by the diverse transportation modes available on land, e.g., *dial and ride-service*, *bike-sharing*, *E-scooter*. Second, we observe that the number of ferries has a strong effect on both excess transit times and departure frequencies. For a high customer service with a departure frequency at the ports of on average every 5 min, a fleet with 20 ferries is required at Kiel fjord. Although the initial acquisition costs for autonomous ferries will be high, operating on water without captain or crew stationed at the ferry likely facilitates the use of more ferries in the long term. Moreover, our experiments show that the same level of service quality can be achieved with 10 fewer ferries if the number of ports in the network is reduced to the most relevant. Third, we observe that more complex time tables with more distinct routes decrease excess transit times by allowing more direct connections in the network. This demonstrates the importance of new technologies providing digital travel planners, allowing passengers to find their journey in an app without the need to study timetables.

Future research should develop a stochastic optimization model for finding the most reliable service network for a fleet of autonomous ferries that may be suspect to bad weather conditions. Another venue is to include the demand pattern into the candidate route generations. With this, one may reduce the number of candidate routes while ensuring that promising routes with direct connections between high demand ports are included. Lastly, one may investigate a combined transportation system, where the fixed schedule service is supplemented by a dial-a-ride service, in which ferries operate on-demand. The dial-a-ride ferries could then be used to serve those ports with less dense demand.

References

1. An, K., Lo, H.K.: Ferry service network design with stochastic demand under user equilibrium flows. Transportation Research Part B: Methodological **66**, 70–89 (2014)
2. Baaj, M.H., Mahmassani, H.S.: An AI-based approach for transit route system planning and design. J. Adv. Transp. **25**(2), 187–209 (1991)

3. Bell, M.G., Pan, J.J., Teye, C., Cheung, K.F., Perera, S.: An entropy maximizing approach to the ferry network design problem. Transp. Res. Part B: Methodol. **132**, 15–28 (2020)
4. Brouer, B.D., Karsten, C.V., Pisinger, D.: Optimization in liner shipping. Ann. Oper. Res. **271**(1), 205–236 (2018). https://doi.org/10.1007/s10479-018-3023-8
5. Cross, J., Meadow, G.: Autonomous ships 101. J. Ocean Technol. **12**, 23–27 (2017)
6. Gu, Y., Góez, J., Guajardo, M., Wallace, S.W.: Autonomous vessels: state of the art and potential opportunities in logistics. NHH Department of Business and Management Science Discussion Paper (2019/6) (2019)
7. Kiliç, F., Gök, M.: A public transit network route generation algorithm. IFAC Proc. Vol. **46**(25), 162–166 (2013)
8. Korsvik, J.E., Fagerholt, K., Laporte, G.: A large neighbourhood search heuristic for ship routing and scheduling with split loads. Comput. Oper. Res. **38**(2), 474–483 (2011)
9. Lai, M., Lo, H.K.: Ferry service network design: optimal fleet size, routing, and scheduling. Transp. Res. Part A: Policy Pract. **38**(4), 305–328 (2004)
10. Lo, H.K., An, K., Lin, W.H., uncertainty: Ferry service network design under demand uncertainty. Transp. Res. Part E: Logist. Transp. Rev. **59**, 48–70 (2013)
11. Maraš, V., Lazić, J., Davidović, T., Mladenović, N.: Routing of barge container ships by mixed-integer programming heuristics. Appl. Soft Comput. **13**(8), 3515–3528 (2013)
12. Ng, M., Lo, H.K.: Robust models for transportation service network design. Transp. Res. Part B: Methodol. **94**, 378–386 (2016)
13. Rolls-Royce: Rolls-Royce and Finferries demonstrate world's first fully autonomous ferry (2018). https://www.rolls-royce.com/media/press-releases/2018/03-12-2018-rr-and-finferries-demonstrate-worlds-first-fully-autonomous-ferry.aspx. Accessed 10 July 2020
14. Thun, K., Andersson, H., Christiansen, M.: Analyzing complex service structures in liner shipping network design. Flexible Serv. Manuf. J. **29**(3–4), 535–552 (2017)
15. Wang, D.Z., Lo, H.K.: Multi-fleet ferry service network design with passenger preferences for differential services. Transp. Res. Part B: Methodol. **42**(9), 798–822 (2008)