INTERNATIONAL
TRANSACTIONS
IN OPERATIONAL
RESEARCH

An approach for efficient ship routing

Gonzalo Romero[a,b], Guillermo Durán[a,c,d], Javier Marenco[e,f] and Andrés Weintraub[a]

[a] *Departamento de Ingeniería Industrial, Universidad de Chile, Chile*
[b] *Sloan School of Management, Massachusetts Institute of Technology, MA, USA*
[c] *Departamento de Matemática and Instituto de Cálculo, FCEN, UBA, Argentina*
[d] *CONICET, Argentina*
[e] *Departamento de Computación, FCEN, UBA, Argentina*
[f] *Instituto de Ciencias, Universidad Nacional de General Sarmiento, Argentina*
E-mail: gromeroy@mit.edu [Romero]; gduran@dii.uchile.cl [Durán]; jmarenco@dc.uba.ar [Marenco];
aweintra@dii.uchile.cl [Weintraub]

Abstract

Ship routing problems are a particular kind of routing problems where the vehicles to be routed are vessels or ships, usually in maritime environments. In contrast to land routing, ship routing has unique features, including overnight trips, disjoint time windows, not necessarily prespecified routes, and a great uncertainty derived from weather conditions. In this work we present a special ship routing problem, which incorporates many features present in general ship routing settings. We discuss aspects related with data gathering and updating, which are particularly difficult in the context of ship routing. Additionally, we present a GRASP algorithm to solve this problem. We apply our solution approach to a salmon feed supplier based in southern Chile, and present computational results on real data.

Keywords: decision support systems; metaheuristics; shipping industry; developing countries

1. Introduction

The "vehicle routing problem" consists of designing efficient routes for a fleet of vehicles, usually with the objective of minimizing operational costs or the number of used vehicles. Several variations of this problem have been extensively studied in the optimization literature, because the efficient routing of vehicles has a great impact in logistic costs.

In contrast, "ship routing" deals with the routing of vessels—usually in maritime environments— and has not received the same attention from the optimization community as the routing of land vehicles. Ship routing has many particular characteristics that must be dealt with, and which are not usually present in vehicle routing problems. Among these characteristics, we highlight the following:

1. Ship voyages may last days, and thus have time horizons for decision-making that, relative to land transport, are both long and of varying duration. This implies, among other things, that overnights at anchor must sometimes be accommodated in the scheduling formulation. Additionally, customers usually have "time windows" where they are available to receive incoming ships. Moreover, because a trip may last several days, we may have to deal with a "set of disjoint time windows" for each customer.
2. The routes assigned to ships are not constrained to follow roads or rails. However, following a straight line between two customers may not always be feasible (e.g., due to the existence of islands or dangerous areas in the sea). Moreover, even in the case where a straight line route between two customers is feasible, it may not be the best route due to opposed winds or currents.
3. As a consequence, determining a route between two customers is not a straightforward task, and two captains may even disagree with respect to the best route. This is particularly critical when the area to be covered has many islands and passages. Due to these facts, the input data for ship routing problems cannot in general be extracted directly from a map, and knowledge from seamen experienced in the routing area must be taken into account.
4. Finally, maritime transport is subject to a great uncertainty because of its relatively high dependence on weather conditions. The estimation of traveling times and costs is a difficult problem compared to land routing, since winds, currents, and weather conditions affect travel times (hence, travel times may not be proportional to travel distances). In extreme cases, a captain can decide to change the route or even to anchor the trip in a safe port due to bad weather.

On the other hand, ship routing problems share some common features with land vehicle routing. For example, maritime fleets are generally not homogeneous, incorporating various types of ships with varying capacities and cost structures, and this situation has also been studied for land routing. Furthermore, time windows and customer priorities are also common features of ship and vehicle routing problems.

The interest in ship routing and maritime transportation in general has been increasingly growing in recent years. Christiansen et al. (2007) gave a recent comprehensive survey about the field, emphasizing the differences between land and maritime routing. The first study of ship routing and scheduling was performed by Ronen (1983). Ten years later, a second paper by the same author (Ronen, 1993) surveyed works on ship scheduling and related topics for the period 1982–1992. Christiansen (2004) also reviewed publications on the subject, concentrating mainly on the years 1993–2003 and including a discussion of the trends expected to impact studies over the decade to come. The latter work reviews exact solution methods and different formulations of the ship routing and scheduling problem.

Concerning the implementation of decision support systems (DSS) for ship routing, Fagerholt and Lindstad (2007) presented such a development for a particular ship routing problem with multiple products and multiple ports. In contrast to this work, the DSS introduced in Fagerholt and Lindstad (2007) does not have the flexibility to decide the amount to be delivered to the customers, and they considered soft precedence constraints. Dauzere-Peres et al. (2007) reported on the implementation of a DSS for the Norwegian company Omya Hustadmarmor that optimizes calcium carbonate deliveries sent by cargo ship from a single plant to paper manufacturers all across Europe. The results obtained using a memetic metaheuristic have enabled company planners

to make faster decisions, boosting both the predictability and flexibility of the supply chain while generating savings in production and shipping costs of about US$7 million a year.

The possibility of accommodating partial deliveries, that is deciding the amount to be delivered to the customer, is an important characteristic of the problem considered in this paper. This is related to the concept of split deliveries in vehicle routing problems, where multiple vehicles are allowed to serve the same customer. This concept was introduced by Dror and Trudeau (1989), and its benefits on routing problems have been studied through several heuristic by various researchers. Archetti and Speranza (2012) gave a recent and comprehensive survey about the field, including both exact and heuristic solution approaches.

This work is organized as follows. In Section 2, we define in detail the particular ship routing problem we are interested in, and we discuss constraints that are frequent in ship routing settings. Section 3 presents our proposed greedy randomized adaptive search procedure (GRASP) for this problem and the addition of an auxiliary knapsack problem for handling partial deliveries. Section 4 provides details on a salmon feed supplier based in southern Chile, which originated this work. Section 5 deals with the issue of determining the input data for the algorithm, in particular with the estimation of routes and travel times among all pairs of customers in a maritime environment, according to the previous comments. Section 6 provides computational results on real data, and Section 7 presents some general conclusions. Finally, Appendices A and B close the paper with details on a DSS implemented for the salmon feed supplier, and details on the sensitivity analysis of the computational results are described in Section 6, respectively.

2. The problem

In this work, we consider a routing problem with time windows enhanced with additional constraints and considerations. We are given a planning horizon of $T \in \mathbb{N}_+$ days, a set C of "customers" to be visited, and a "distance matrix" $D \in \mathbb{R}^{|C| \times |C|}$ such that d_{ij} specifies the distance separating customer i from customer j, for $i, j \in C$. Each customer is located by the sea and has a port capable of receiving incoming ships. Each customer $i \in C$ has an associated demand $d_i \in \mathbb{R}_+$ of a good, which we assume can be arbitrarily split, and an associated due date $t_i \in \{1, \ldots, T\}$, which denotes the day where the customer is expecting the delivery. We are also given a set S of heterogeneous "ships." For each ship $s \in S$, we have a particular capacity $c_i \in \mathbb{R}_+$, an average speed $s_i \in \mathbb{R}_+$, a fixed cost per day $fc_i \in \mathbb{R}_+$, and a (variable) cost per traveled nautical mile $vc_i \in \mathbb{R}_+$.

A "feasible solution" consists of a set of "routes" involving distinct ships. Each route is defined by a ship from S and a sequence $(c_1, q_1), \ldots, (c_k, q_k)$ such that $c_i \in C$ and $0 < q_i \leq d_i$ for $i = 1, \ldots, k$. The sequence $c_1, \ldots, c_k$ determines the customers to be visited by the ship, and the quantity q_i represents the amount to be delivered to customer c_i. The routes must, additionally, satisfy the following constraints:

1. The (individual) capacity of each ship cannot be exceeded. This is a hard constraint due to safety requirements. As usual in routing problems, the capacity of the ships is expressed in the same units as the customers demands.

2. Some customers cannot receive every ship in the fleet, due to ship sizes and machinery. This compatibility constraint arises for customers with small ports and/or limited docking and unloading facilities, and is not usually present in land routing problems.

3. Each customer has soft time windows, specifying a disjoint set of periods in which the customer can receive incoming ships. These time windows correspond to the usual constraints in vehicle routing (e.g., a customer can be delivered from 8 a.m to 1 p.m). However, since the ship routes may last several days, these constraints generate a set of disjoint time windows for each customer. More concretely, if a ship arrives at a customer at a time not included in the time windows, then it must wait until the dock opens at the start of the next working day before unloading. The customer $i \in C$ has an associated "late delivery penalty" $ldp_i \in \mathbb{R}_+$ per unit of time, incurred when the ship arrives to the customer on a day later than the due date t_i.

4. Each customer has an associated priority. A customer cannot be preceded in the same route by another customer with lower priority. There are typically a few categories, associated with urgent delivery status and, more specifically to maritime routing, due to biosecurity risk classifications. Biosecurity issues arise in customers with bad health records, or with waters containing viruses or pollution. A ship visiting such a customer automatically gets into a risky condition, and cannot visit a "healthy" customer afterwards.

5. All sailing ships should be loaded at least at p percent of their capacity in order to make their trip profitable, where p is a fixed parameter for each instance. If a ship $i \in S$ departs from the port with less than p percent of its capacity c_i, then a "minimum load penalty" $mlp_i \in \mathbb{R}_+$ is incurred in the objective function.

6. Finally, complete demand of a customer may not be delivered, but the customer may receive a smaller quantity on the due date. In order to maintain a reasonable service level, a parameter d is associated with each instance, and each customer must receive at least d percent of their demand. However, partial deliveries should be avoided, as they negatively affect the service level provided to the customer. To model this negative impact, we introduce for each customer $i \in C$ a "partial delivery penalty" $pdp_i \in \mathbb{R}_+$ incurred when the customer receives less than their original demand, namely d_i units.

The first four constraint types—namely, capacity, compatibility, soft time windows, and priorities—are already present in the literature on maritime routing, although is not common to consider all of them together. The last two are new, to the best of our knowledge. They originated from the operations of a salmon feed supplier in southern Chile, described in Section 4, which motivated this paper.

   For now, let us point out that the minimum load penalty models a soft constraint motivated by the mechanism by which the ship owners are being paid, as the salmon feed supplier does not own the fleet. In Section 4, we consider $p = 80\%$, as negotiated by the company and the ship owners. On the other hand, the partial delivery penalty is partly motivated by the fact that each partial delivery, besides reducing the service level provided to the customer, generates an urgent delivery for the following day. This in the context of a rolling horizon schema, where the problem described in this section is solved daily with an horizon of $T = 3$ days, but only the first day decisions are implemented. Therefore, any demand not delivered on its due date is added to the instance being solved the next day, assigning to it the highest possible priority. A more detailed description of partial deliveries and the rolling horizon schema are provided in Sections 4 and 6, respectively.

Similarly, more details concerning the biosecurity risk classifications, described in constraint 4, in the context of the salmon industry in Chile are provided in Section 4.

The "objective function" considered in this problem combines fixed costs, variable costs, and penalties for poor customer service quality. If $B \subseteq S$ is the set of ships used in the feasible solution and $UB \subseteq B$ is the set of underloaded ships, then the objective function can be written as

$$F = \sum_{i \in B} (fc_i \, \text{days}_i + vc_i \, \text{dist}_i) + \sum_{i \in UB} mlp_i \sum_{j \in C} (ldp_j \, \text{late}_j + pdp_j \, \text{partial}_j),$$

where days_i represents the number of days in the horizon that ship $i \in S$ is used, dist_i represents the total traveled distance for ship $i \in S$, late_j is a binary indicator specifying whether the delivery to customer $j \in C$ is late or not, and partial_j is a binary indicator specifying whether the delivery to customer $j \in C$ is partial or not. The problem asks for a feasible solution minimizing the objective function. For example, in the operations of a salmon feed supplier in southern Chile, described in Section 4, each customer's demand may represent the amount of food needed for several days, therefore in principle it is possible to deliver only the quantity required for the first day, as long as the remainder is delivered in the following day. This problem is clearly NP-hard, since it contains the classical vehicle routing problem as a subproblem.

3. A GRASP-based algorithm

Considering that an integer linear programing model of a simplified version of the ship routing problem described in Section 2 presented unacceptable solution times for a typical daily scheduling process, a heuristic approach was undertaken, opting for a version of GRASP. We chose such an approach for its relatively simple computational implementation, as well as its record of good results with similar problems to the present one. We refer the reader to Romero (2008) for the complete integer programing formulation and the computational results.

3.1. Literature review on GRASP

The GRASP (Feo and Resende, 1995; Festa and Resende, 2002) is a multistart metaheuristic frequently used in combinatorial problems. In GRASP, each iteration consists of two phases: construction and local search.

In the construction phase, an initial feasible solution is built iteratively in a greedy randomized manner. Namely, at every step the next decision is selected at random from the set of the best possible local decisions. This is followed by a local search phase, where the neighborhood of the initial solution is explored, changing to a new solution each time the evaluation of a neighbor solution shows it is better than the current one. The procedure continues until all neighbors of the current solution have equal or worse performance. In this manner a local minimum is arrived at, and the best solution among all the local optima is returned as the final solution.

GRASP has been successfully applied in a wide array of situations ranging from vehicle routing and scheduling, facility location and telecommunication to biology, logic, partitioning and

assignment problems, as well as graph theory. A recent review of GRASP that embraces both methodological aspects and applications in the literature since 1989 is found in Festa and Resende (2009a, 2009b). Probably, the most relevant application of GRASP in relation to our work is by Kontoravdis and Bard (1995), where a vehicle routing problem with time windows is formulated and solved using GRASP, with the objective of minimizing the number of vehicles required and the total distance traveled. They assume, however, a homogeneous fleet of vehicles. To the best of our knowledge, this work presents the first application of GRASP to a ship routing problem.

3.2. Overall schema

Algorithm 1 provides a general description of the proposed GRASP procedure. Each iteration involves the construction of an initial feasible solution with Algorithm 2 (described in Section 3.3), followed by a local search with Algorithm 3 (described in Section 3.4). These two procedures are executed taking just d percent of the demand for every customer, in order to maximize the number of served customers. However, partial deliveries should be avoided, therefore the obtained solution is subject to a knapsack-based procedure (described in Section 3.5), which maximizes the number of completely served customers in the obtained route. Finally, the postoptimization backtracking procedure given by Algorithm 4 (described in Section 3.6) is applied to the resulting solution, in order to find an optimal visiting sequence for the customers in the route. The procedure is repeated until an iteration limit or a time limit is reached, and it returns the best solution found.

Algorithm 1. Overall GRASP procedure

$\bar{s} :=$ empty solution;
$i := 1$ //Iteration number
while $i \leq$ maximum number of iterations **and** time limit not reached **do**
   Construct a feasible solution s_1 considering d percent of the demand for each customer (Algorithm 2);
   Apply local search to s_1, obtaining s_2 (Algorithm 3);
   Complete as many customers as possible in s_2 with the knapsack-based procedure described in Section 3.5, obtaining a new solution s_3;
   Apply backtracking to s_3, obtaining s_4 (Algorithm 4);
   if $F(s_4) < F(\bar{s})$ **then**
      $\bar{s} := s_4$;
   end if
   $i := i + 1$;
end while
Return $\bar{s}$;

3.3. Generation of an initial feasible solution

Initial feasible solutions are generated using the random greedy procedure described in Algorithm 2, which takes two parameters $k \in \mathbb{N}$ and $q \in \mathbb{N}$. The solution is built iteratively. First, a ship is randomly selected from the first k ones on the ship list ordered by decreasing capacity. If $k = 1$

then the ship selection is deterministic, and if $k = |S|$ then the ship selection if fully random, so an intermediate value seems to be the best choice. In the experiments presented in Section 6, the value $k = 3$ was assumed, and this value provided satisfactory results. Then, customers are assigned to it until its capacity is reached.

Algorithm 2. Generation of an initial feasible solution

$C :=$ set of customers;
$S :=$ list of ships in decreasing order of capacity;
while $C \neq \emptyset$ **do**
    $s :=$ randomly chosen ship from the first k ships in S;
    $C_s :=$ customers from C compatible with s;
    $E :=$ set of the q customers in C_s further away from the port;
    Assign to s a randomly chosen customer $c \in E$;
    Update C_s;
    $C := C \backslash \{c\}$;
    while $C_s \neq \emptyset$ and the capacity of s is not exceeded **do**
       $D :=$ set of the q customers in C_s nearest to the center of mass of the customers already assigned
       to s;
       Assign to s a randomly chosen customer $c \in D$;
       Update C_s;
       $C := C \backslash \{c\}$;
    end while
    $S := S \backslash \{s\}$;
end while

Customer assignment is done by randomly choosing a customer from the list of customers who are compatible with the vessel. Compatibility is determined by the accessibility (docking and unloading) constraints, and the condition that the minimum delivery quantity defined for the customer is less than the ship remaining capacity. Clearly, customers who have already been assigned to a ship in the current solution are considered incompatible with every vessel. In order to generate a reasonable schedule, only the q customers closest to the center of mass of the customers previously assigned to the vessel are included in this list. Again, different values of q generate different solutions that range from greedy to random. In the experiments described in Section 6 the value $q = 5$ was taken, obtaining satisfactory results.

To start the customer assignment procedure, the first customer is randomly chosen among those further away from the port, according to the travel distances discussed in detail in Section 5. The intuition behind this selection is to try to avoid the situation where customers who are very far away from the port, and from each other, are left for the last vessel to serve. Each time a new customer is selected, the remaining ship capacity value is updated, which in turn redefines the list of compatible customers. It is worth noting that this procedure is only based on the geographic locations of the customers. Moreover, it does not give a special consideration to customers priorities when generating the initial feasible solutions. We resort to the objective function within the local search described in the next section in order to handle the customer priorities.

The customer assignment procedure is repeated until there are no more customers compatible with the ship under consideration. This will occur when remaining capacity is insufficient, all remaining customers are not accessible by the ship, or all customers have already been assigned.

Finally, the overall procedure stops when all salmon customers have been assigned, and then the local search phase starts. If some customers cannot be assigned to any ship, the GRASP iteration terminates and the algorithm proceeds to construct the initial solution of the next iteration.

Note that the technique of randomly choosing a customer from the bounded list, that is, not necessarily selecting the best one, allows GRASP to generate good quality initial solutions that are different at each iteration.

To avoid possible infeasibilities in the generation of initial solutions, fictitious ships are incorporated into the problem. These vessels have very high fixed and variable costs that discourage their use unless absolutely necessary, and thus will naturally tend not to have customers assigned to them in the local optimal solutions found by the local search procedure.

Finally, it should be noticed that the problem of recognizing whether an instance is infeasible, given a certain configuration of feed demand orders and available ships, is not trivial. The inclusion of fictitious ships is thus a simple technique for finding solutions such that if an instance turns out to be infeasible, a large number of orders will still be scheduled.

3.4. Local search

Algorithm 3 presents the local search procedure applied to all the feasible solutions constructed by Algorithm 2. The local search procedure of the algorithm utilizes the two most common acceptance criteria in VRPTW (Bräysy and Gendreau, 2005) contexts: accept either the best neighbor in the entire neighborhood, if it is better than the current solution, or the first neighbor that improves on the current solution. Each of these criteria was applied to one-half of the initial solutions generated. These two criteria are integrated in lines 2 and 5 of Algorithm 3, by stating "best/first neighbor," a decision we parameterize in the algorithm.

The behavior of each acceptance criteria was tested over instances of the heterogeneous vehicle routing problem published by Taillard (1996). These tests showed us that, although using the best neighbor criterion gives more stable and on average superior results, applying the first improvement criterion can lead to better objective function values related to the diversification of the search. The strategy of using each acceptance criterion for one-half of the searches was adopted as a diversification device, on the basis that the solutions could be stabilized with the best neighbor rule, without losing the opportunity to include less-frequented solutions using the first improvement rule.

Algorithm 3. Local search procedure, enhanced with a parameter specifying whether the best neighbor in the neighborhood or the first neighbor that improves on the current solution is considered.

$s :=$ solution generated by Algorithm 2;
$\bar{s} :=$ best/first CROSS neighbor of s w.r.t. the objective function;
while $s \neq \bar{s}$ **do**
    $s := \bar{s}$
    $\bar{s} :=$ best/first CROSS neighbor of s w.r.t. the objective function;

end while
for each route r in s **do**
   $C_r :=$ set of customers in r.
   Replace r by the best ordering of C_r (satisfying the priorities) w.r.t. the objective function;
end for

For each solution, the neighborhood is created using the CROSS exchange procedure between two routes defined by Taillard et al. (1997), for the vehicle routing problem with time windows. An example is shown in Fig. 1. The procedure begins by eliminating arcs $(i-1, i)$ and $(k, k+1)$ from one route, and arcs $(j-1, j)$ and $(l, l+1)$ from the second route. The segments $i-k$ and $j-l$, which may include an arbitrary number of customers, are then exchanged between the two routes by adding new arcs $(i-1, j)$ and $(l, k+1)$ to the first route and $(j-1, i)$ and $(k, l+1)$ to the second. Proving that a given solution is locally optimal, in a CROSS neighborhood for a particular pair of routes, involves an $O(n^4)$ procedure.

The local search process includes all route pairs in the initial feasible solution for exploration. A pair is selected at random, and the CROSS neighbors are evaluated until the acceptance criterion is met. If a better solution is found compared to the current one, the algorithm is updated and the process is reinitiated. The iteration terminates when no better neighbor is found, after all possible pair of routes have been considered.

When the local search ends, using either of the two acceptance criteria, a local optimum has been found. Moreover, it would be very difficult to improve on it without creating a completely different configuration. In other words, no trivial change between any pair of routes would improve the solution as valued by the objective function. This is the case because the CROSS neighborhood is very large, and every route pair is explicitly assessed.

Evaluating the objective function of a neighbor of the current solution only involves recalculating the costs and penalties for the two routes modified by the CROSS exchange. However, evaluating it exactly, that is considering the best possible sequence of the customers included in each route, can still be expensive. This is due to the large number of neighbors assessed. To address this, when evaluating the objective function of a route, we perform a 2-opt procedure in order to search for a good sequence of the customers included in the route of each ship. This 2-opt procedure is a simple variation of the classical 2-opt local search, modified in order to take into account the precedence constraints given by the customer priorities. This heuristic procedure generates a good quality customer visits configuration for the ship, in a relatively short time and is therefore used in each evaluation of the objective function.
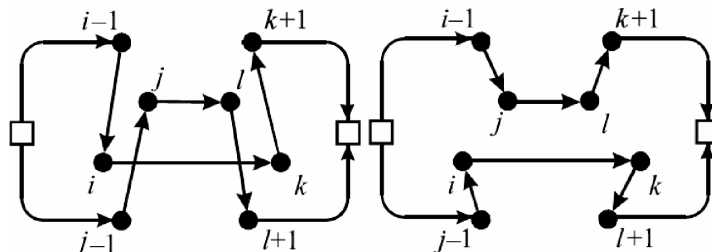

Fig. 1. CROSS exchange.

3.5. Handling partial deliveries

An important aspect of the ship routing problem considered in this work is the possibility to deliver each customer just a fraction of their demand. Recall that we require the final schedule to deliver at least d percent of the demand to each customer, and the remainder is left to the following day as an urgent delivery (and this is why partial deliveries should be avoided). This particular characteristic adds flexibility to the solution process, since we can choose to partially serve a customer, which if completely served would not fit within the total capacity of the ship.

Handling partial deliveries within a heuristic procedure is not straightforward, since each customer can be delivered any amount above the d percent of its demand. Such continuous decisions can be naturally handled within a mixed-integer programing model via continuous variables, but—as already mentioned—such an approach is not viable for the instance sizes we faced in this application. In order to overcome this difficulty, we implemented the following simple procedure which produced adequate results.

The execution of Algorithms 2 and 3 takes as input data just d percent of the original demands (i.e., the minimum amount that can be delivered to the customers). The solution generated by Algorithm 3 contains, therefore, partial deliveries to all the customers. In order to complete as many customers as possible, we solve for each ship a continuous knapsack problem with knapsack capacity equal to the ship remaining capacity. We consider one knapsack item per customer, with weight equal the undelivered demand and benefit normalized to one (recall that our objective is to maximize the number of complete deliveries). This postprocessing step does not add a significant amount to the total running time, since the continuous knapsack problem can be solved in linear time. Moreover, according to our computational experience, it contributed to generate solutions with a good number of complete deliveries.

3.6. Postoptimization phase

The final solution is subjected to a postoptimization procedure using backtracking to define the optimal visit sequence for each route $\bar{R}$. To this end, Algorithm 4 is applied to every route r in the solution. This recursive algorithm takes as input a set T of customers and a partial route R. In the initial call of the recursive procedure, R is an empty route and T is the set of customers in $\bar{R}$. The algorithm also maintains a global variable BR with the best route found so far.

Algorithm 4. Recursive backtracking procedure that finds the optimal sequence of customers for a given route. This algorithm takes as input a set T of customers to be considered and a partial route R to be completed.

for each $c \in T$ **do**
    $R' := R \cup \{c\}$, where customer c is added at the end of the route R.
    $T' := T \setminus \{c\}$;
    if $T' = \emptyset$ **and** $F(R) < F(BR)$ **then**
        $BR := R$;
    end if
    if $T' \neq \emptyset$ **then**

Recursively call this algorithm with R' and T';
end if
end for

4. Application to a salmon feed supplier

We applied the proposed GRASP algorithm within the implementation of a DSS for scheduling salmon feed deliveries for Skretting Chile S.A. This company supplies nutrition products and services for sustainable salmon farming in southern Chile. The feed is transported by ship from the feed producer warehouse to its salmon farm customers.

One of the main activities of the salmon farming industry is the cultivation of the salmon in cages kept at farms in the sea for 16–18 months. This process occurs after an initial stage in which the salmon is raised as fingerlings in fresh water. The fish must be fed regularly at the farms with a range of inputs depending on their stage of development, and the desired quality of the final product.

For salmon producers attempting to fulfill product quality targets, a fundamental requirement is the delivery of fish feed supplies in the necessary quantities, at the right time and place. Unreliable feed supply leads to increased expense, due to the additional time needed for the fish to reach the ideal harvesting weight, which in turn imposes an opportunity cost in terms of the delay in starting the next batch of salmon. In general terms, feed is the most significant expense item in the salmon production process, accounting for approximately 70% of total production costs.

The distribution operations of the feed producer are triggered by the orders placed by its clients. These orders are detailed with the salmon farm, feed type, and delivery date requested, and have to be placed at least two weeks before the delivery date, in order to have enough time to produce and distribute the feed. Moreover, unless a fatal accident caused the whole production of a farm to be replaced by fish with different dietary necessities, demand orders are firm and are assumed to be given in the mathematical model presented in the previous sections.

The maritime transport operations are handled jointly by the feed producer and a local shipping firm, through the former private port installations at Pargua, in the south of Chile. The feed producer pays the shipper a set rate per tonne, for deliveries to each salmon farm. Additionally, the feed producer decides the ship routing and scheduling of the vessels that will deliver the feed. The two companies decide the actual rate and the number of vessels that will participate in the feed distribution operations in twice yearly negotiations.

Under this contract structure, once the salmon farm customer demand is known the short-term transport costs are fully determined and the routing decisions reduce to finding a configuration whose operational costs are acceptable to the shipping firm. In particular, the shipping firm enforces that all used ships must have a minimum of 80% of its capacity loaded. However, in the medium term, the feed producer will attempt to minimize actual shipping costs, including the fixed cost associated to the use of the vessels, in order to lower the contracted cost, by sharing the potential cost reductions between the two parties in future rate negotiations. In particular, if the number of required vessels can be reduced, there are significant fixed costs savings associated. Supporting the ship routing decision-making process, with aim of minimizing its fixed and operating costs while still having a high service level to its customers is thus the main purpose of the ship routing and scheduling DSS described here.

The feed producer transport operations revolve around the work of the company transport planner, whose job is to decide which feed orders are assigned to which available ships, and in what sequence the deliveries are to be made. In other words, he or she is responsible for determining the route of each ship, avoiding whenever possible late or incomplete deliveries as is discussed below, at the lowest operating cost. The assignment is performed manually, requiring approximately 2 hours per day, and depends heavily on the planner's experience in balancing the length of the routes and the ship cargo load levels.

There are no set rules for the decision-making process but the following are among the criteria considered:

- Ships should leave the port with the highest percentage of used capacity possible.
- The more distant the destination, the larger the assigned ship so that unit costs per ton are minimized.
- Certain physical restrictions must be observed, such as not sending large ships to farms with small warehouses that do not have the facilities to handle them.

In defining the sequence of the customers visited by each ship, the planner must also consider the biosecurity issues faced by the salmon farming industry. In 2007, a crisis erupted as the infectious salmon anemia (ISA) virus spread rapidly through the majority of seawater farming centers, dealing a serious blow to the sector development (Godoy et al., 2007). As the ISA virus management strategies being implemented when the DSS reported here was under development, the biosecurity risk level of the various farms had to be taken into account. The four possible biosecurity levels, in increasing risk order, are free, suspicious, quarantine, and outbreak. These levels were defined by a Chilean government agency, and are used in the industry to classify each salmon farm. A ship that has visited a farm with a given risk level, cannot then call at one with a lower risk classification without first being subjected to a disinfection process. The disinfection process can only be carried out at a major port, for the feed producer this means returning to its base port at Pargua.

Travel times between the port and the various salmon farms range from 1 to 13 hours, under reasonable weather conditions. The majority of the farming operations have small storage facilities, with a capacity of approximately 80 tonnes, and so tend to place frequent feed orders, requiring an average of three visits per week to be served. The orders placed by the customers almost never exceed the capacity of their warehouses, and therefore in the mathematical model, we assume the demand to be less than the storage capacity. Note also that most of the farm operations are set up to receive deliveries only between 8 a.m and 6 p.m. Finally, another complicating factor in determining optimal ship assignment is that the shipping firm's fleet varies in actual operating costs (as opposed to contract costs) and capacity, the latter characteristic ranging uniformly from 95 tonnes for the smallest vessels to 210 tonnes for the largest.

The transport planner schedules two to six ships every day, each one visiting between four and eight salmon farm operations. A complete daily schedule will therefore involve an average of four vessels and 24 farms. These daily averages are the lower bound of the problem size since in general the planning horizon is more than one day. Additionally, demand and available ship capacity rarely coincide in practice. Typically, some loads are delayed while others are moved up to achieve better capacity utilization rates. In cases where demand exceeds available distribution capacity, the planner will determine a schedule that guarantees partial deliveries of at least $d = 60\%$ of each

farm's demand, thus ensuring no farm is left without feed for an entire day. This policy results in delivery shortfalls that are redefined as late orders, with an urgent priority level for delivery over the following day, and must therefore be scheduled for the start of the route they are assigned to, adding a further complication to the scheduling process.

There are three criteria for deciding whether a farm will receive partial, or full delivery on a given day, when ship capacity is less than demand. First, those nearest the base port are prime candidates for partial delivery, given that ships pass close to them every day, enabling overdue deliveries to be made without a major detour. Second, farms located in areas where demand for deliveries the next day is high are also chosen for partial delivery. Third, farms that are difficult to add to a route, due to their geographical locations, or because they have a virus contamination problem that implies precedence restrictions, will receive full delivery on the requested date if possible.

In the case where demand is less than ship capacity, but not so much less that the deliveries can be made using one less ship, the planner will use the excess capacity to load orders intended for the following day. These will be delivered on the return trip, thus generating a longer than normal route given that the vessel will have to drop anchor for the night and continue deliveries the next day. In such situations, the ship will have to seek out a protected location, particularly if rough weather conditions, or a storm, is expected.

The solutions found by the GRASP procedure strongly depend on the costs and penalties associated to each of the terms in the objective function. The greatest values were assigned to the fixed cost fc_i for every ship $i \in S$, which gives the cost for each day the ship i is operating in the solution, and to the ship underload penalty mlp_i for $i \in S$. Next, in decreasing order of value, are the late delivery ldp_j and partial delivery pdp_j terms for each customer $j \in C$, both related to the quality of customer service the producer wishes to provide. The smallest cost is applied to fuel expense (i.e., the coefficient vc_i for $i \in S$), which is a function of fuel prices.

Finally, let us point out that the salmon feed production and distribution operations in Chile look amenable for a vendor managed inventory (VMI) strategy (Simchi-Levi et al., 2003), where the feed supplier decides the time and amount of all the orders subject to explicit service level requirements. However, the salmon industry in Chile seems not to be mature enough for this kind of contracts, in the sense that the lack of trust and information sharing among the different actors of the supply chain is still very high. This industry characteristic was made explicit during the damage control stage of the ISA virus discussed in Section 6 of this work.

5. Estimation of travel distances

A major task in the application of automatic routing techniques to maritime environments is the estimation of the distances between the customers. The application described in the previous section involves more than 200 salmon farms, and no systematic data were previously available. Furthermore, the Euclidean distance separating any pair of farms was often a poor estimate of the actual travel distance. This is due to the highly complex local geography with its countless islands, straits, and sandbars. Actually, in our case, the ships serving the farms tend to follow the coastline rather than sail in a straight line.

In this section, we propose a distance estimation methodology that was useful in this particular application, and we believe that this may be of interest in similar settings.
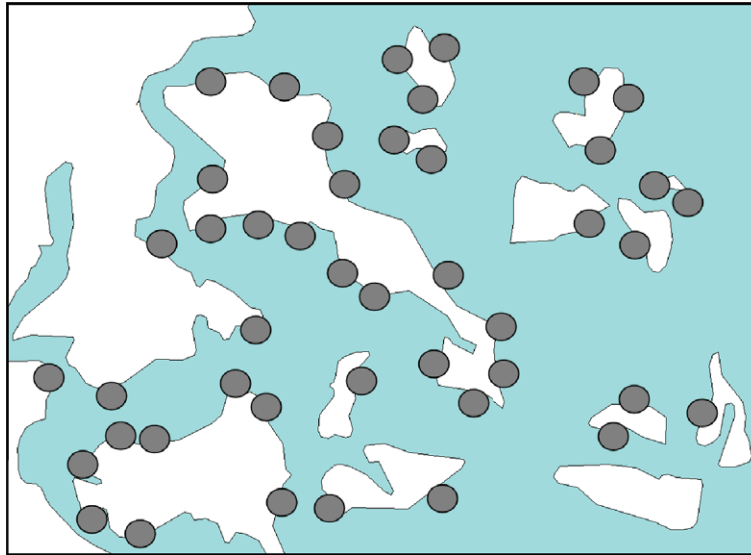
Fig. 2. Chiloe Island salmon farm customers—latitude: −42.5°, longitude: −73.5°.

The estimation starts by defining a reduced network, which is a representative of the complete network in that all of the arc lengths in the latter can be estimated from those in the reduced version with only a small margin of error. To arrive at this definition, the salmon farm locations and the geography of the area served by the company were studied. Then, a subset of arcs was identified that allowed all of the distances in the matrix to be represented as compositions of the subset members. This reduced network contains approximately 2500 arcs, or 13% of the total number of arcs in the complete network. Moreover, it only considers arcs between nearby farms.

The next step was to measure explicitly the arc lengths in the reduced network. This was accomplished using Google Earth. Information from interviews with ships' captains was used to ascertain the most frequently used routes between given salmon farms, among the many possible alternatives.

Finally, the Floyd-Warshall algorithm (Floyd, 1962; Warshall, 1962) was applied to the reduced network to estimate the shortest route between every pair of farms. This generated more than 20,000 length data that were taken as estimates of the real distances. A somewhat similar approach was followed by Fagerholt et al. (2000), where they estimate the distances between ports as shortest paths in a network that models the land contours by nodes and arcs that represented obstacles, or infeasible sailing areas.

To illustrate, Fig. 2 shows a subset of salmon farm customers in and around Chiloe Island, part of the region served by the company. In Fig. 3, the same customers are grouped into areas, and so-called "pivot farms" belonging to two or more areas are indicated by a different tag color. Instead of explicitly measuring the distances between every pair of farms in Fig. 2, we only measure the distance between farms in the same area. The distance between farms that are further away is estimated as a composition of the measured arc lengths, and thus necessarily include one or more pivot farms in the path.
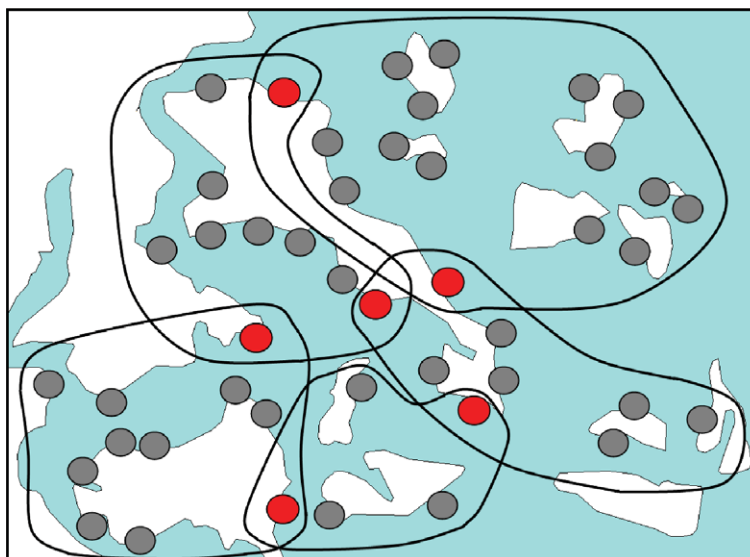
Fig. 3. Chiloe Island areas and pivot farms—latitude: $-42.5°$, longitude: $-73.5°$.

We tested our methodology by comparing our estimates to the route distances calculated explicitly using Google Earth. This comparison included origin destination pairs from a set that was considered particularly difficult to estimate by the feed company, and also for origin destination pairs chosen at random. We found that the differences between our estimates, and the distances measured explicitly, averaged less than 3%, a margin of error considered reasonable by the firm. Additionally, data from actual navigation routes were collected and used as a benchmark, obtaining similar results.

Travel time between two salmon farms is defined by their distance, and the average speed of the ship serving the route. Finally, a fixed percentage of this computation is added as a simple approach to provide a safety margin against travel time uncertainty. Comparisons between the estimated travel times, and the company's actual shipping operations showed that, as long as marine weather conditions were not excessively adverse, the estimated times were very close to those actually observed, attaining a 4% average error.

6. Computational results

In this section, we present the results of the tests conducted to evaluate the performance of the DSS for maritime shipment scheduling described in the previous sections. The tests were based on 10 days of actual company operating data for the period June 16–25, 2008.

The weights for the terms in the objective function described in Section 2 were decided after a process of calibration, and in consultation with the feed producer. We used a rolling horizon, where three days were evaluated in each execution of the prototype, and only the first day decisions were implemented. This procedure was followed trying to avoid myopic solutions. There was a 20-minute

limit imposed by the company to solve the daily instances. This time included inputting data and entering any posterior manual modifications. Therefore, the actual time allowed for generating solutions was no more than 10 minutes. As it turned out, the solution times for each execution during testing never exceeded the 10-minute limit; indeed, they averaged about 5 minutes, well within the company's requirements.

The turnaround time for ships returning to port was taken to be 6 hours in the prototype, which is the average time required for carrying out reverse logistics operations, biosecurity disinfection and loading of new feed orders in actual operations. Additionally, the minimum load percentage for ships was set at $p = 80\%$, while the minimum percentage of demand to be delivered to each customer was set at $d = 60\%$. Both of these values are in accordance with the company criteria, as described in Section 4. Additionally, we performed sensitivity analysis on the values of parameters p and d. The results of these analysis are summarized at the end of this section in Section 6.2, while the details are presented in Appendix B.

The tests performed compared the schedules generated by the prototype, with those designed by the producer's transport planners. Any later changes due to bad weather or coordination problems with the company's production side were not taken into account. In other words, comparisons were confined to the schedules, as opposed to their implementation, evaluated in terms of the objective function described in Section 2.

It was thus assumed that the scheduling was implemented as originally planned. We acknowledge that this leads to a possible overestimation of the benefits generated by the ship routing and scheduling DSS. In practice, some of the opportunities identified in the planning process by the prototype might be lost due to adverse weather conditions or lack of coordination with the production side or the customers.

However, during the testing period, no significant changes took place in the scheduling designed by the company. Moreover, when major difficulties do arise in the transportation operations, the transport planner generally makes only local modifications on certain routes, without conducting a reevaluation of the problem as a whole to generate a new schedule. This practice may change with the implementation of the DSS proposed here, given that it can readily execute such reevaluations on the basis of updated information.

The main results for the two sets of data and the gap between them are summarized in Table 1. The variables compared are the distance traveled, the planned ship days, the number of farms scheduled for late and partial delivery, the tonnage delivered, and the nautical miles per tonne delivered.

Table 1
Summary of results

	Actual	Prototype	Gap (%)
Distance (nautical miles)	2,743.73	2,568.27	−6.39
Ship days	45	36	−20
Late deliveries	20	7	−65
Partial deliveries	6	0	−100
Tonnes shipped	3,247.25	3,278.87	0.97
Nautical miles/tonne	0.84	0.78	−7.30

First of all, note that the total ship travel distance in the prototype schedules was 6.39% shorter than that from the company schedules. This result can be surprising, as this factor was not the only criterion in the objective function, nor even the most significant one. Moreover, the ships in the prototype schedules carried almost 1% more cargo. This slight load increase is explained by the fact that, with the prototype, it was necessary to include the ability to move up deliveries from the day after the end of the 10 days horizon. In fact, without this, infeasibilities or illogical solutions would be proposed in order to satisfy the 80% minimum load percentage constraint at the end of the horizon. That is, we allowed the prototype to consider demands from June 26, 2008 in order to satisfy the minimum load constraint. Needless to say, requiring that deliveries pending at the end of the planning horizon are zero is not representative of actual operations, which are continuous.

A more eloquent measure of the difference between the prototype and company's actual schedules is the number of nautical miles traveled per tonne transported. On this indicator the gap in favor of the prototype scheduling was 7.3%, a significant improvement in our view given that nautical miles per tonne is a secondary objective of the DSS schedules, according to the objective function discussed at the end of Section 2.

Additionally, the quality of service induced by the schedules was an even more important performance measure in this case. The prototype displayed a large improvement in this area, as indicated by the 65% decline in order volumes scheduled for late delivery, and the complete absence of partial deliveries, even though there was flexibility to schedule them. Though the benefit to the feed producer of an increase in service quality would be difficult to quantify, we note that the company has identified this factor as a key area in its efforts to attain a major competitive advantage in the market in which it operates.

Furthermore, by using a three-day rolling horizon the prototype scheduling procedure was able to anticipate problems generated by orders from salmon farms in distant locations or infected with the ISA virus, thus achieving a more efficient utilization of the fleet of ships. In fact, it achieved a 20% reduction in ship days used to make all deliveries in the planning horizon. In fact, the improvement on this factor was the principal objective of the DSS. This benchmark was set by the feed producer at the beginning of this project. Moreover, if the number of ships required for its delivery operations can be consistently lowered, then the company will be in a position to negotiate lower shipping rates with the shipping firm. This potential for cutting costs in the medium term is very significant, and considerably outweighs the possible short-term savings generated by further reductions in the variable costs associated with nautical miles traveled. Additionally, let us note that a further benefit of using fewer ships is the decrease in damage to the environment.

Finally, it should be noted that the drop in nautical miles obtained with the prototype schedules could be enlarged by reducing service quality, that is, by raising the number of late or partial shipments. To do this, the current weights on the objective function terms would have to be adjusted so as to increase the relative importance of travel distance.

6.1. Daily scheduling results

The complete daily results of the schedules generated by the prototype, the actual company scheduling and the gaps between them are set out in Table 2. The objective function terms shown are the

Table 2
Comparison of prototype and actual company schedule results

Day	June 16	June 17	June 18	June 19	June 20	June 21	June 22	June 23	June 24	June 25
					Company					
Distance	527.91	361.65	94.34	403.11	203.92	393.47		301.36	348.36	109.61
Ship days	8	6	2	7	4	6		4	6	2
Late deliveries	4	3	0	4	0	1		3	5	0
Partial deliveries	1	0	1	1	0	2		1	0	0
Tonnes	695	327.5	210	431.25	311.75	455		330.5	403.75	82.5
					Prototype					
Distance	473.66	460.32	140.48	324.5	218.29	400.39		278.02	272.61	
Ship days	6	6	3	5	3	6		4	3	
Late deliveries	2	2	0	0	0	1		0	2	
Partial deliveries	0	0	0	0	0	0		0	0	
Tonnes	700	462.8	195	393.75	308	605		341.82	272.5	
Solution time (s)	530	165	217	366	77	401		174	229	
					Gap					
Distance (%)	−10.28	27.28	48.91	−19.50	7.05	1.76		−7.74	−21.74	
Ship days (%)	−25.00	0.00	50.00	−28.57	−25.00	0.00		0.00	−50.00	
Late deliveries	(2)	(1)	–	(4)	–	–		(3)	(3)	
Partial deliveries	(1)	–	(1)	(1)	–	(2)		(1)	–	
Tonnes (%)	0.72	41.31	−7.14	−8.70	−1.20	32.97		3.43	−32.51	

same as those given in Table 1, with the addition of the prototype solution times in seconds, which as noted in the main text never exceeded 10 minutes and averaged about half that time.

As can be seen, the prototype completes the entire delivery schedule for the 10-day planning horizon with one day to spare. This means that no deliveries are made on the last day (June 25). This was due mainly to the fact that the model schedules fewer ship-days and thus tends to have more ships available as the planning horizon progresses. This in turn generates more options for delivering full orders on time.

Note also that the runs scheduled by the prototype are not always shorter than the actual company runs. In fact, with a three-day moving horizon, the system can anticipate difficult situations, such as deliveries to distant customers, or farms infected by the ISA virus and generate schedules that may be locally worse, but are globally superior over a longer horizon.

6.2. Sensitivity analysis

In the results previously described, we considered the minimum load percentage for ships to be $p = 80\%$, while the minimum percentage of demand to be delivered to each customer was set at $d = 60\%$. While these values are in accordance with the company criteria, it is important to explore how sensitive are the results to different values of these parameters.

Table 3
Gap with respect to the transport planner solution

	$p = 100\%$ $d = 60\%$	$p = 60\%$ $d = 60\%$	$p = 80\%$ $d = 80\%$	$p = 80\%$ $d = 40\%$
Distance (nautical miles, %)	13.96	−9.26	4.53	−3.48
Ship days (%)	−6.67	−20	−20	−20
Late deliveries (%)	30	−70	−35	−70
Partial deliveries (%)	150	−83	−67	−67
Tonnes shipped (%)	1.66	.65	1.41	3.3
Naut miles/tonne (%)	14.39	−9.07	3.27	−6.55

Therefore, we replicated the previously described experiment considering different values for d and p. In Appendix B we present representative results, where for the minimum load percentage for ships we considered the values $p = 100\%$ and $p = 60\%$. Similarly, for the minimum percentage of demand to be delivered to each customer, we considered $d = 80\%$ and $d = 40\%$. For each of these cases we present tables similar to Table 1, that provide a summary of the performance of the solution generated by the DSS. In Table 3, we present the relative gaps of all the metrics considered in Table 1, with respect to the transport planner schedule for each of these experiments.

Considering larger values for p restricts the flexibility on the minimum percentage of the ship capacity that must be loaded on each trip, therefore the DSS generates lower quality solutions. In particular, when $p = 100\%$ each ship must be completely loaded, and the DSS generates a schedule that is of considerable lower quality than the base case of $p = 80\%$. Moreover, the quality of the solution generated is lower than the one designed manually by the transport planner. In fact, while the schedule generated by the DSS reduces the number of ship days needed, it increases every other metric, including nautical miles traveled per tonne delivered, and the number of partial and late deliveries. This suggests that the flexibility of allowing routes where ships are not fully loaded is a fundamental characteristic of our algorithm in order to achieve good results.

On the other hand, considering smaller values for p increases this flexibility, and we should expect potentially better solutions from the DSS. However, when $p = 60\%$ the schedule generated by the DSS achieves very similar performance compared to the base case of $p = 80\%$, on each metric considered. Additionally, the average solution time of the DSS increases by about 50% as shown in Table B5, probably due to the expanded feasible region that the algorithm is exploring. This suggests that the DSS could be used to further explore the potential of achieving better solutions by setting smaller values of p, with the objective of providing quantitative support in the negotiation process between the company and the ship owners.

Similarly, considering larger values for d restricts the flexibility on the minimum demand percentage that has to be served to each customer, therefore the DSS generates lower quality solutions. In particular, when $d = 80\%$ the schedule generated by our algorithm achieves a much poorer performance in terms of the nautical miles traveled per tonne delivered and the number of late deliveries compared to the base case of $d = 60\%$. Moreover, its performance in the former metric is worse than the schedule generated by the transport planner, as shown in Table 3. However, let us emphasize that the schedule proposed by the DSS needs the same number of ship-days than the base case, which is the most important metric for the company. This suggests that a larger

value of d decreases the quality of the solution proposed by the DSS mainly in the secondary objectives.

Finally, considering smaller values for d increases the feasible region that the algorithm is exploring, and therefore we would expect better quality solutions compared to the base case of $d = 60\%$, while at the same time incurring in larger solution times. In particular, when $d = 40\%$ the schedule generated by the DSS achieves a very similar performance compared to the base case of $d = 60\%$, on each metric considered. However, the solution time increases in more than two times compared to the base case, as shown in Table B5. Moreover, the solution times exceeded the 20-minute limit for each daily solution imposed by the company in many instances, as shown in Table B6. This suggests that the potential of achieving better solutions by setting smaller values of d is small in this instance of the problem, and it does not justify the large increase in the solution time.

7. Conclusions

This work presented a framework to deal with ship routing problems, including the gathering of travel times and the handling of several particular constraints that arise in maritime environments. We applied the proposed methodology to the scheduling of ship routes for a salmon feed supplier in southern Chile, obtaining interesting results.

Due to the size of the considered instances (which arise from real operational data), an integer programing approach was not practical in this case. Moreover, the final objective of this work was the implementation of a DSS for the company, and the running times of a state-of-the-art integer programing solver for these instances was unacceptable for a daily scheduling. A GRASP-based algorithm was proposed and implemented, and the results were satisfactory. The proposed algorithm handled the particular constraints associated with ship routing quite well, including partial deliveries and priorities among the customers. In particular, the knapsack algorithm for dealing with partial deliveries was adequate, and we consider that such an approach may be useful in similar settings.

A major issue in ship routing problems is the estimation of travel distances. The methodology in this work to deal with this problem can be applied in similar situations, specially when the geography of the considered area of operation includes islands or makes sailing by the coastline mandatory. The routes among the customers and the resulting travel times estimated by the DSS were approved by the ship captains, and reasonably approximated real routes and times. This is a very important feature in this context, since automatically generated trips in maritime environments are more prone to contain errors, or underestimations, than land trips following roads. Therefore, rejection of the schedules by the captains is a potential problem.

Finally, it must be noted that the uncertainties in travel times due to weather conditions and maritime currents may have a huge impact in the real performance of the planned trips. We addressed this issue by adding a security margin to the travel times. However, this particular characteristic of ship routing (which is also present in land routing although with a lesser degree of uncertainty) certainly deserves further study. The usual currents, dominating winds, and weather forecasts for the routing area may be taken into account in order to generate a robust schedule, and it would be interesting to study how to combine these factors into a scenario-based optimization approach.

Acknowledgments

References

Archetti, C., Speranza, M.G., 2012. Vehicle routing problems with split deliveries. *International Transactions in Operational Research* 19, 1–2, 3–22.

Bräysy, O., Gendreau, M., 2005. Vehicle routing problem with time windows, Part I: route construction and local search algorithms. *Transportation Science* 39, 1, 104–118.

Christiansen, M., Fagerholt, K., Nygreen, B., Ronen, D., 2007. Maritime transportation. In Barnhart, C. and Laporte, G., (eds) *Handbook in OR & MS*, Vol. 14, Elsevier, Amsterdam, pp. 189–284.

Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship routing and scheduling: Status and perspectives. *Transportation Science* 38, 1, 1–18.

Dauzere-Peres, S., Nordli, A., Olstad, A., Haugen, K., Koester, U., Myrstad, P.O., Teistklub, G., Reistad, A., 2007. Omya hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufacturers. *Interfaces* 37, 1, 39–51.

Dror, M., Trudeau, P., 1989. Savings by split delivery routing. *Transportation Science* 23, 141–145.

Fagerholt, K., Heimdal, S., Loktu, A., 2000. Shortest path in the presence of obstacles: an application to ocean shipping. *Journal of the Operational Research Society* 51, 683–688.

Fagerholt, K., Lindstad, H., 2007. Turborouter: an interactive optimisation-based decision support system for ship routing and scheduling. *Maritime Economics & Logistics* 9, 214–233.

Feo, T., Resende, M., 1995. Greedy randomized adaptive search procedures, *Journal of Global Optimization* 6, 109–133.

Festa, P., Resende, M., 2002. GRASP: an annotated bibliography. In Ribeiro, C., Hansen, P., (eds) *Essays and Surveys on Metaheuristics*. Kluwer Academic Publishers, Norwell, MA, pp. 325–367.

Festa, P., Resende, M., 2009a. An annotated bibliography of GRASP – Part I: Algorithms. *International Transactions in Operational Research* 16, 1, 1–24.

Festa, P., Resende, M., 2009b. An annotated bibliography of GRASP – Part II: Applications. *International Transactions in Operational Research* 16, 2, 131–172.

Floyd, R.W., 1962. Algorithm 97: shortest path. *Communications of the ACM* 5, 6, 345.

Godoy, M., Kibenge, F., Aedo, A., Kibenge, M., Groman, D., Grothusen, H., Lisperguer, A., Calbucura, M., Avenda no, F., Imilan, M., Jarpa, M., 2007. Primera detección, aislamiento y caracterización molecular de ISA-V en salmón del atlántico (salmón salar) de cultivo en Chile (in Spanish). *SalmoCiencia Online* 3, 47–55.

Kontoravdis, G., Bard, J., 1995. A grasp for the vehicle routing problem with time windows. *ORSA Journal on Computing* 7, 1, 10–23.

Romero, G., 2008. Sistema de apoyo a las decisiones de ruteo maritimo para un proveedor de alimentos para salmones (in Spanish), Degree thesis, Departamento de Ingeniería Industrial, Universidad de Chile.

Ronen, D., 1983. Cargo ships routing and scheduling: survey of models and problems. *European Journal of Operations Research* 12, 119–126.

Ronen, D., 1993. Ship scheduling: the last decade. *European Journal of Operations Research* 71, 3, 325–333.

Simchi-Levi, D., Kaminsky, P., Simchi-Levi, E., 2003. *Designing & Managing the Supply Chain*. McGraw Hill, New Delhi.

Taillard, E., 1996. A heuristic column generation method for the heterogeneus fleet VRP. Technical Report CRT-96-03, Centre for Research on Transportation, University of Montreal, Canada.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., Potvin, J.I., 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 31, 2, 170–186.

Warshall, S., 1962. A theorem on Boolean matrices. *Journal of the ACM* 9, 1, 11–12.

Appendix A: The decision support system

In this appendix, we provide details on the decision support system (DSS) implemented for Skretting Chile S.A.

A.1. Characterization of the salmon farm customers

The data for defining customers orders are their name, latitude and longitude of customers' location used to build the interfaces and to compute the Euclidian distances between them, and their demand, priority and minimum delivery percentage, the latter to be set by the final user. Their earliest and latest delivery dates might coincide if no flexibility is allowed. And the start and end time of their working day that will define their delivery time window.
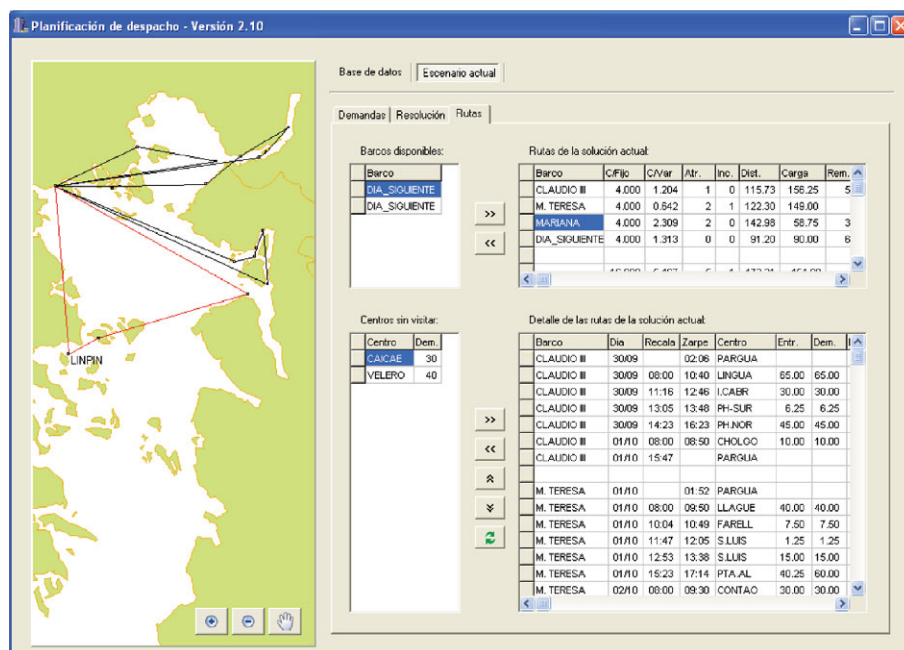

Fig. A1. The prototype interface.

A.2. Insertion of new salmon farms

A characteristic of the salmon industry that is accommodated by our DSS is the frequent entry of new salmon farms into the company's network. This occurs every three weeks on average as new farms are opened by its customers, or existing ones change their location or switch to the company from a different feed supplier.

It was therefore necessary to develop an efficient method of inserting new farms into the customer network. In the method devised, at most four new arcs are added to the existing reduced network to connect the new farm, and the shortest routes between all the farms in the reduced network thus modified are recalculated to obtain new estimates of the real distances.

To define a new arc, all the existing farms are positioned in relation to the lines of latitude and longitude passing through the location of the new farm, and in each of the four quadrants so formed the farm closest to the new one, according to the Euclidean distance, is selected (if any exists). The user then inputs the corresponding real distances calculated using Google Earth or nautical charts, which become the definitive distance data for the new reduced network. To insert the new farm in the original reduced network, a new area is created consisting of the new farm and the four closest ones selected as just explained. These four farms will serve as the pivots linking the new area to the previously existing ones.

This method for inserting new customers into the network was utilized during development of the DSS prototype to add approximately 20 new farms, thus providing an opportunity to test the method usefulness and efficiency. The time required to update the distance matrix is only a few seconds and thus the time needed to input a new farm will depend exclusively on how long it takes the user to make the necessary distance measurements, which can be no more than four arc lengths, one for each quadrant. Comparisons of the distances estimated using this approach with those obtained through explicit measurement found errors of less than 7% on average, a figure considered to be reasonable by the feed producer.

The increase in error level over the percentage obtained for the original reduced network (less than 3%) occurred mainly because the latter was defined using human criteria based on the geographical and nautical characteristics of each farm whereas the later modifications incorporated only Euclidean distance data. But the fact that adding a new farm only requires that the user explicitly measures a maximum of four distances is adequate compensation for the greater degree of error. Since we would reasonably expect the errors to increase over time as new farms are added, the distance estimates generated by the method would have to be corrected periodically. This would be done through a study similar to the one described here initially in which additional criteria other than distance were incorporated, with the goal of bringing the average error back down to the 3% range.

A.3. Characterization of the ships

The relevant data for describing the ships are their name, capacity, average speed, fuel consumption, and delivery unloading rate. Also their daily fixed cost, which must be the real cost that will not necessarily coincide with the contract figure, to be used in the calculations for minimizing the number of ships utilized. In fact, as it was mentioned in Section 4, the feed producer interest is to

minimize the real operational costs as opposed to the contract cost and particularly the number of vessels necessary to run the distribution of salmon feed to its clients.

Finally, considering that each ship starts being available at a particular time in the horizon, as opposed to assuming that all of them are available at the beginning, contributes to a more realistic modeling of the company's operations and allows the evaluation of rolling time horizons of more than one day. In fact, the number of available vessels dynamically changes in time as vessels come back from previous assignments or are scheduled to be released from scheduled maintenance.

A.4. DSS prototype development

The heuristic and the prototype interfaces were all developed in the Borland C++ Builder 5 environment. The prototype interface features tabs for selecting panes where different tasks can be carried out. There are panes for inputting and modifying permanent data on salmon farm characteristics and daily data on delivery orders, and also for accessing optimization and solution evaluation functions where schedule planners can incorporate their practical experience, test possible modifications, and evaluate their consequences.

More specifically, a demand tab displays a pane for inputting data on customer orders to be delivered that are drawn from Excel forms maintained by sales personnel. The data are automatically aggregated by salmon farm and delivery date (ignoring the detailed product characteristics) to reduce the complexity of the problem, a map is generated indicating the farm locations and the time window limits are calculated.

A solution tab opens a pane for modifying the parameters of the objective function that evaluates the solutions and the number of initial solutions for the GRASP optimization. Changes to this parameter will vary the execution time and the breadth of the search for solutions. The solutions generated can be stored for later retrieval and export to an Excel data grid in a standard format.

Finally, in a routes pane, planners can manually modify a generated solution on the basis of their practical experience and understanding of specific operating situations that are not modeled in the prototype (see Fig. A1). Changes can be made to evaluate each farm's position on a route, the ship a farm is assigned to and the quantity of feed to be delivered, as well as the effects of these changes on the objective function terms and the estimated arrival and departure times at the port.

An additional benefit of the prototype is its ability to quickly evaluate the impact on each objective function term of potential changes in route planning requested by customers. This makes it an important support tool for the feed producer when negotiating last-minute schedule modifications. The ability to automatically calculate the arrival and departure times for each salmon farm as part of the evaluation of route alterations is also a valuable function for the company since it substantially improves the estimated delivery time information provided to customers when such changes are made.

A.5. Remarks on the implementation

The prototype was tested using actual company data for a 10-day operating horizon, generating a reduction of 7.3% in nautical miles traveled per tonne of feed delivered. It also achieved a major increase in service quality, lowering the number of late deliveries by 65% and eliminating partial

deliveries altogether. Additionally, the solution provided by the prototype generated a 20% decline in the number of ship-days used for completing the deliveries scheduled within the planning horizon.

Among other benefits demonstrated by the prototype, particularly notable is its ability to generate good quality solutions in a matter of minutes, in contrast to the various hours required by the producer's transport planner using manual procedures. The implementation of the tool would thus free up company resources for application to other value-generating activities. It would also allow the planner to quickly analyze a range of alternative scenarios and produce higher quality and potentially more robust solutions than those obtained with the traditional methods.

A further advantage of the prototype is that through its objective function, actual operating costs are explicitly analyzed. This affords the producer new support tools for negotiating transport contracts with the shipping firm. It also allows the company to make speedy evaluations of the impact of potential scheduling changes requested by customers and then negotiate them on the basis of accurate estimates of the cost implications for both sides.

A major intangible benefit was the systematization of the data on salmon farm distances (for creating the distance matrix) and on ship characteristics. With this information, estimates could be made of the arrival and departure times at each salmon farm for a given route with only a small error margin. This capability was exploited early on in the development of the prototype to improve the quality of the information provided to company customers.

The definitive implementation of the support system was postponed due to the collapse of the salmon farming industry that occurred in Chile in late 2007 and early 2008 with the rapid spread of the ISA virus. Poor management of the biosecurity threat coupled with the unusually rapid growth of the industry enabled the virus to contaminate more than 37% of Chile's 600 salmon farms belonging to 25 different companies. Before the crisis, the industry had become the country third largest exporter after copper and wood pulp, posting international sales of more than US$2 billion in 2008.

A recent diagnosis of the salmon farming sector commissioned by financial institutions indicates that the industry will need at least one more year to return to the levels of productivity required for profitability. The companies themselves have stated publicly that they are unlikely to get back to pre-crisis production volumes before the end of 2012. Due to the high mortality and slow growth rates caused by the virus, farm operators were forced to move up the date of the 2008 harvest and seeding came to a halt. The result was a fall in production levels from a record high of 600,000 tonnes in 2008 to 450,000 in 2009 and 2010, and 400,000 tonnes during 2011. Employment in the industry has also been hit, falling from 45,000 workers in the industry in 2008 to about 25,000 workers in 2011.

Naturally, the demand for salmon feed fell drastically as well and the company was forced to reduce its costs. The operations manager who was the main driving force behind the project in the firm was laid off, and at the same time, along with the drastic reduction in the production volumes, the size of the ship routing and scheduling problem decreased significantly, and thus the manual methodology to solve it became attractive again.

Efforts are currently underway to establish a new legal and operating framework for the industry that will help to prevent further crises in the future. And it is expected that in the medium term the salmon industry in Chile will recover the growth rate it had exhibited in the last two decades before the crisis.

Appendix B: Sensitivity analysis results

In this section, we present tables with the details of the sensitivity analysis performed on the computational results from Section 6. In Section 6, we considered the minimum load percentage for ships to be $p = 80\%$, while the minimum percentage of demand to be delivered to each customer was set at $d = 60\%$. In this section, we present the results obtained by replicating the experiment, but now considering $p = 100\%$ and $p = 60\%$, and $d = 80\%$ and $d = 40\%$.

Tables B1, B2, B3, and B4, we summarize the performance of the schedule proposed by the DSS on the metrics discussed in Section 6 for each value of p and d considered. Additionally, they present the relative gap with respect to the schedule designed manually by the transport planner. In Table B5, we present the relative gap of the average solution time with respect to the base case. Finally, in Table B6 we present the performance of the daily schedules proposed by the DSS, together with the solution times in each case.

Table B1
Results with $p = 100\%$, $d = 60\%$

	Actual	Prototype	Gap (%)
Distance (nautical miles)	2,743.73	3,189	13.96
Ship days	45	42	−6.67
Late deliveries	20	26	30
Partial deliveries	6	15	150
Tonnes shipped	3,247.25	3,302	1.66
Naut miles/tonne	0.84	0.97	14.30

Table B2
Results with $p = 60\%$, $d = 60\%$

	Actual	Prototype	Gap (%)
Distance (nautical miles)	2,743.73	2,511.14	9.26
Ship days	45	36	−20
Late deliveries	20	6	−70
Partial deliveries	6	1	−83.33
Tonnes shipped	3,247.25	3,268.58	0.65
Naut ical miles/tonne	0.84	0.77	−9.07

Table B3
Results with $p = 80\%$, $d = 80\%$

	Actual	Prototype	Gap (%)
Distance (nautical miles)	2,743.73	2,873.97	4.53
Ship days	45	36	−20
Late deliveries	20	13	−35
Partial deliveries	6	2	−67
Tonnes shipped	3,247.25	3,293.58	1.41
Nautical miles/tonne	0.84	0.87	3.27

Table B4
Results with $p = 80\%$, $d = 40\%$

	Actual	Prototype	Gap (%)
Distance (nautical miles)	2,743.73	2,651.49	−3.48
Ship days	45	36	−20
Late deliveries	20	6	−70
Partial deliveries	6	2	−67
Tonnes shipped	3,247.25	3,358.08	3.3
Nautical miles/tonne	0.84	0.79	−6.55

Table B5
Gap with respect to the base case $p = 80\%$, $d = 60\%$

	$p = 100\%$	$p = 60\%$	$d = 80\%$	$d = 40\%$
Average solution time (%)	79.34	55.86	−13.11	232.93

Table B6
Prototype schedule results for different parameter values

$p = 100\%$, $d = 60\%$

Day	June 16	June 17	June 18	June 19	June 20	June 21	June 22	June 23	June 24	June 25
Distance	704.18	350.61	269.01	449.18	194.2	577.43		135.39	509	
Ship days	8	6	2	7	2	8		2	7	
Late deliveries	1	3	5	10	1	3		0	3	
Partial deliveries	4	3	1	0	2	2		1	2	
Tonnes	757	409	160	388	230	715		178	465	
Solution time (s)	633	651	273	90	608	620		48	949	

$p = 60\%$, $d = 60\%$

Day	June 16	June 17	June 18	June 19	June 20	June 21	June 22	June 23	June 24	June 25
Distance	546.39	194.02	278.07	281.72	279.33	387.42		213.91	330.28	
Ship days	7	3	5	3	4	6		3	5	
Late deliveries	1	1	0	0	1	0		0	3	
Partial deliveries	1	0	0	0	0	0		0	0	
Tonnes	735.25	266	273.75	321.25	386.25	575.5		255.58	455	
Solution time (s)	536	131	261	193	875	372		256	741	

$p = 80\%$, $d = 80\%$

Day	June 16	June 17	June 18	June 19	June 20	June 21	June 22	June 23	June 24	June 25
Distance	632.61	187.22	306.69	352	227.44	485.28		138.31	544.38	
Ship days	7	2	4	4	3	8		2	6	
Late deliveries	2	2	0	2	0	2		0	5	
Partial deliveries	1	0	0	0	1	0		0	0	
Tonnes	745	218.75	392.5	326.25	298	718.08		177.5	417.5	
Solution time (s)	210	428	269	78	335	286		41	229	

Table B6
Continued

$p = 80\%, d = 40\%$

Day	June 16	June 17	June 18	June 19	June 20	June 21	June 22	June 23	June 24	June 25
Distance	585.97	288.1	156.12	318.24	326.1	443.48		111.86	421.62	
Ship days	7	3	5	4	4	7		1	5	
Late deliveries	1	1	1	0	0	1		0	2	
Partial deliveries	1	0	0	1	0	0		0	0	
Tonnes	735	332.5	310	357.5	367.5	671.83		112.5	471.25	
Solution time (s)	899	1123	696	193	1203	643		86	2345	