Math. Appl. **6** (2017), 127–141
DOI: 10.13164/ma.2017.08

DOWNSTREAM LOGISTICS OPTIMIZATION AT EWOS NORWAY

MARTIN BRANDA, KJETIL K. HAUGEN, JAN NOVOTNÝ AND ASMUND OLSTAD

Abstract. The Norwegian company EWOS AS produces fish feed for the salmon farming industry, supplying approximately 300 customers spread along the coast of Norway. The feed is produced at three factory locations and distributed by a fleet of 10 dedicated vessels. The high seasonality of the demand and the large number of customers make the distribution planning a substantial challenge. EWOS handles it by operating a system of mostly fixed routes with decentralized planning at each factory. The distribution can be described as a multi-depot vehicle routing problem with time windows, multiple vehicle usage, inter-depot routes, heterogeneous fleet and a rolling horizon. The paper presents a mathematical model for this problem, which is solved by heuristics and meta heuristics. Based on detailed historical data collected by EWOS during the autumn of 2010, the model has proposed a dynamic set of routes with a significant reduction of travelled distance – close to 30% – and an increase of average vessel fill-rate – from 60% up to 95%. This implies a substantial fuel saving, with a positive environmental impact, and also a potential for down-scaling the fleet, with additional considerable cost savings for the company.

1. Introduction

This paper presents an application of optimization techniques to a real-world routing problem, which, given its characteristics, ranges among the most advanced vehicle routing problems. The problem appears at the company EWOS AS, which supplies a wide portfolio of aqua feed products for the international aquaculture industry. The Norwegian branch of the company has three production plants situated along the west and north coasts of Norway in Florø, Halsa and Bergneset. Its customer base consists of local fish farms spread along the Norwegian coastline. The company has a heterogeneous fleet of vessels that serve the customer demand. Individual customer orders are normally known at least 10 days in advance. However, customers are allowed to change order amounts (within reason) up to vessel departure time.

The demand profile exhibits seasonality with high peak in summer, which is preconditioned by the biological life cycle of the fish. The distribution planning is

---

MSC (2010): primary 90B06; secondary 90C59.

Keywords: vehicle routing problem, multi depot, time windows, inter-depot routes, multiple vehicle usage.

The authors gratefully acknowledge that this project was partly supported by The Research Council of Norway, P.O Box 564 N-1327 Lysaker, Norway. The project was also supported by EWOS AS. One of the authors would also like to express his gratitude to prof. A. Løkettangen (in memoriam), who introduced him to the tabu search technique.

a task with rolling horizon where the fleet operates on fixed routes and customer orders define the filling of the vessels.

The characteristics of the distribution problem as present in EWOS are described in Section 3, a formal mathematical model is presented in Section 4 and the solution methods are discussed in Section 5.

During autumn 2010, the company was collecting their operational data, including customer orders and the exact delivery routes of their vessels. This data made it possible to carry out a very accurate comparison where we let the algorithm overtake the route planning at given moments of the history. The results are presented and discussed in Section 6.

2. Literature review

Transportation optimization problems were introduced more than 50 years ago [12]. Many papers and monographs have been published on the topic since then. Vehicle routing problems (VRPs) generally involve an assignment of vehicles to trips in order to minimize travel costs. This section presents sources relevant to the EWOS distribution problem, i.e., papers on variants of VRP and solution techniques, and survey papers which provide classifications of VRPs and literature overviews. The listing is by no means exhaustive.

An excellent introduction to VRP, its variants and existing solution procedures can be found in [31]. Also [4, 5, 18] contain chapters relevant for the problem. [8] gives a detailed review and classification of maritime transport models – namely strategic decision models focusing on fleet composition and tactical and operational planning models dealing with scheduling of deliveries and travel-cost minimization for short-term problems. In [20] a survey can be found of sources on road and maritime transportation and fleet composition for various variants of VRP including time windows, heterogeneous fleet and multiple depots. [24] presents a recent literature survey on the fleet size and mix problem in maritime transportation. An extensive comparison of different solution methods for VRP with heterogeneous fleet is presented in [3].

Due to the complexity of the VRPs, heuristics are commonly used to solve real-sized problem instances. Across the literature, we can discern several basic heuristics classes. Constructive heuristics build routes by sequentially adding customers to a partially constructed feasible solution. Two-phase heuristics perform clustering of customers before or after route construction. Improvement heuristics, e.g. inter-route exchanges, can be applied to each of the cases, see [21]. Insertion heuristic variants arise as a result of the decision about which unrouted customer to pick and where to insert it into the partial solution in each iteration. Clustering relates to identifying the groups of customers to be served within the same area at approximately the same time. Each cluster is then served by one vehicle which reduces the problem to a set of independent instances of the classical travelling salesman problem (TSP), see [19]. In the early paper [30], various heuristics for solving time-constrained VRPs were considered, and the best performance was reported with insertion heuristics. The efficiency of the insertion methods was reported proportional to the number of time window constraints and their tightness. This result is due to the lower number of possible feasible insertions in such problems. However, this paper dealt only with homogeneous fleets. [23] considered the combined fleet-design, ship-scheduling and cargo-routing problem with limited availability of ships and proposed a clustering procedure together with a genetic algorithm. [7] presented a hybrid genetic algorithm for the multi-trip vehicle routing problem, in which each vehicle can perform several trips during the working day. [2] introduced adaptive large neighbourhood search for the vehicle routing problem consisting in determining the routing of a fleet of vehicles when each vehicle can perform multiple routes during its operations period. [26] proposed a mathematical formulation for the multi-depot vehicle routing problem with heterogeneous vehicles which was solved by a variable neighbourhood search algorithm. [32] developed two meta-heuristics based on neighbourhood exploration for multi-depot vehicle fleet mix problems. [27] used a heuristic and an ant-colony meta-heuristic in their decision support system to generate routes over city transportation network and incorporated detailed individual vehicle routes to Google Maps.

There are approaches which combine heuristic and exact optimization techniques. [14] developed a three-phase routing strategy for VRP with heterogeneous fleet, multiple depots and time windows. The first phase aims at identifying a set of cost-wise effective feasible clusters, the second phase assigns clusters to vehicles and sequences them on each tour. Ordering of nodes within clusters and scheduling of vehicle arrival times at the nodes is performed in the third phase. In [22], feasible routes are generated first, then a reduction algorithm based on dominance rules is applied to reduce the number of feasible routes and finally the routes for each ship are selected based on solving an exact optimization problem. The tabu search technique, introduced by Glover [16], has been highly successful when applied to the VRP, see [11]. Evolutionary algorithms, i.e. genetic and memetic algorithms, were also used to solve various VRP.

In the delivery problem of EWOS, the customers decide about placing the orders independently. Another strategy, so called vendor-managed inventory, allows the supplier to choose the timing and size of deliveries, ensuring that the customers do not run out of stock. Such problems are called inventory routing problems, see e.g. [8, 28, 29]. There is an increase in papers considering stochastic aspects of logistic planning, see [25] for a review.

3. Problem description

The downstream distribution ship planning in EWOS is currently done manually and within a system of fixed routes, where current customer orders define the filling of vessels before leaving the factory. Usually, each route visits many customers where the duration can range from a few hours up to several days. Because of high seasonal demand variations in customer demands, the (manually set) fixed routes normally vary between the peak (summer) season and off-peak (winter) season. The distribution is decentralized so that each factory is responsible for the vessel transportation to cover the demand in their specified customer region. EWOS are fully aware of the fact that their distribution planning can be improved by using an operations research methodology, dynamic routes and centralized planning where one planning unit is responsible for covering all the company customers' demand from the three company-owned factories. The goal of this project is to analyze the possibilities for improvements in fill rates of vessels by utilizing a VRP methodology.

The mathematical model formulation must take into consideration the following aspects. Ships usually visit many customers and perform continuous routes with varying travel time. One single ship can perform multiple trips during the planning horizon – returning to the depot, loading new products and going for another trip. For each customer order, there is the earliest and latest possible delivery time specified. The model must, therefore, incorporate time windows. EWOS supplies a large variety of products, however, there is no real limit as to the number of product kinds loaded on a vessel because they can be transported in bags. The EWOS fleet is heterogeneous in terms of vessel size and travelling speed. Naturally, he travel costs are different for each vessel, too. Orders must always be satisfied with a single complete delivery (i.e. no delivery splitting is allowed). The three factories of EWOS play the role of depots in the VRP, hence the problem is a multi-depot one. EWOS requested that the model shoould allow for inter-depot routes. Loading and unloading times are significant compared to travelling times and they are proportional to the cargo amount. Dynamic scheduling principle takes into account that some ships can be still operating at the beginning of the planning horizon, i.e., not all ships are available at once and we only know the time when a ship will arrive to the depot, thus, it will be available for loading. Considering this, routes optimization can be started at any time when the orders for the following planning horizon are known. We can hardly find any paper which considers all of the above mentioned aspects together.

The problem under consideration is NP hard – it is a generalization of VRP with time windows which is itself a generalization of TSP with time windows. See [15] for computational complexity and transformations between problems.

4. MIP model

In this section we present the distribution problem of EWOS formulated as a mixed integer [33] linear program. The model can be described as a multi-depot vehicle arc routing model with delivery time windows, heterogeneous fleet, and multiple inter-depot routes. The time in the formulation is continuous (not discretized). We use the following *index sets*:

- $\mathcal{V}$ – vessels ν,
- $\mathcal{J}$ – factory visits j,
- $\mathcal{Q}$ – customer orders q,
- $\mathcal{Q}_\nu$ – customer orders which can be satisfied by vessels ν,
- $\mathcal{J} \cup \mathcal{Q} = \mathcal{N}$,
- $\mathcal{J} \cup \mathcal{Q}_\nu = \mathcal{N}_\nu$ – nodes feasible for vessels ν,
- $\mathcal{N}_\nu^o = \mathcal{N}_\nu \cup \{o(\nu)\}$, $\mathcal{N}_\nu^d = \mathcal{N}_\nu \cup \{d(\nu)\}$, $\mathcal{N}_\nu^{od} = \mathcal{N}_\nu \cup \{o(\nu), d(\nu)\}$.

Elements q of the set $\mathcal{Q}$ represent the individual customer orders. Two different orders may be placed by the same customer. The customer locations are reflected in the $T_{nn'\nu}$ time-distance coefficients defined below. The ordered quantity is denoted by D_q and the requested time-window is given by T_q^{min} and T_q^{max}.

Elements of the set $\mathcal{J}$ represent visits of the vessels to the factories. As opposed to the set $\mathcal{Q}$, which is given, the set $\mathcal{J}$ must be pregenerated. That is, for each physical factory, a number of visit elements must be created. The set should be big enough to cover the expected number of visits (in our case $|\mathcal{J}| \approx 30$). The factory corresponding to each $j \in \mathcal{J}$ can be again distinguished by the time-distance parameter $T_{nn'\nu}$. The model formulation is such that not all of the visit nodes available must necessarily be used.

The initial and final vessel positions are denoted by $o(\nu)$ and $d(\nu)$, respectively, and can be regarded as geographical positions. When dealing with routing sequences, we use the term **node** and employ the indices n or n' for orders q, factory-visits j and any of $o(\nu)$ or $d(\nu)$. A visit to node n is also called a *service*, meaning the unloading of goods at a customer or a vessel loading at a factory.

We use the following decision *variables*:

- $x_{nn'\nu} \in \{0,1\}, n \in \mathcal{N}_\nu^o, n' \in \mathcal{N}_\nu^d, \nu \in \mathcal{V}$ – whether vessel ν services node n' directly after node n or not,
- $t_n \geq 0, n \in \mathcal{N}_\nu^{od}$ – the time at which service at node n starts,
- $y_{\nu n} \geq 0, n \in \mathcal{N}_\nu^o, \nu \in \mathcal{V}$ – on-board inventory of vessel ν after servicing node n,
- $a_{\nu n} \geq 0, n \in \mathcal{N}_\nu^o, \nu \in \mathcal{V}$ – amount loaded/unloaded to/from ship ν at node n.

The *parameters* that come into the model are:

- $c_{nn'\nu}$ – cost of vessel ν going from n to n',
- D_q – requested quantity in customer order q,
- U_n – (un)loading (service) time per product unit at node n,
- K_ν – capacity of vessel ν,
- T – time horizon,
- $T_{nn'\nu}$ – time distance between service nodes n and n' for vessel ν,
- T_n^{min}, T_n^{max} – time window for service n, $T_n^{max} \leq T$,
- $o(\nu)$ – start node of vessel ν,
- $d(\nu)$ – end node of vessel ν.

The cost parameter $c_{nn'\nu}$ can be set equal to the travel distances or to the consumed fuel amount, provided that the vessel fuel consumptions are known. The time horizon T represents the planning period and imposes an upper bound on the visiting times. Unless $T_n^{min} < T$ for each n, the order must be disregarded or the planning period prolonged. We are now ready to give an exact mathematical formulation of the operational planning problem. A discussion of possible extensions accompanies the formulation.

4.1. Objective function

$$\min \sum_{\nu \in \mathcal{V}} \sum_{n \in \mathcal{N}_\nu^o} \sum_{n' \in \mathcal{N}_\nu^d} c_{nn'\nu} x_{nn'\nu} + \alpha \sum_{\nu \in \mathcal{V}} t_{d(\nu)}. \tag{4.1}$$

The objective function (4.1) seeks to minimize the sum of transportation costs. This is the first term in the objective. The second term, which may be viewed as a penalty, is an approximation of the true cost structure involved in ships not finishing their routes in due time. Establishing this true cost structure turned out to be a too complex task in this setting, so for practical usability, a parameter α allows the user to add parametric weight on this part. By adjusting the α coefficient, the times at which ships finish their routes may be given appropriate importance.

4.2. Vessel flow constraints

$$\sum_{\nu\in\mathcal{V}}\sum_{n\in\mathcal{N}_\nu^o} x_{nj\nu} \leq 1, \ j \in \mathcal{J}, \tag{4.2}$$

$$\sum_{\nu\in\mathcal{V}}\sum_{n\in\mathcal{N}_\nu^o} x_{nq\nu} = 1, \ q \in \mathcal{Q}_\nu, \tag{4.3}$$

$$\sum_{n\in\mathcal{N}_\nu} x_{o(\nu)n\nu} = 1, \ \nu \in \mathcal{V}, \ n \in \mathcal{N}_\nu, \tag{4.4}$$

$$\sum_{n\in\mathcal{N}_\nu^o} x_{nn'\nu} = \sum_{n\in\mathcal{N}_\nu^d} x_{n'n\nu}, \ \nu \in \mathcal{V}, \ n' \in \mathcal{N}_\nu, \tag{4.5}$$

$$\sum_{j\in\mathcal{J}} x_{jd(\nu)\nu} = 1, \ \nu \in \mathcal{V}, \tag{4.6}$$

$$\sum_{q\in\mathcal{Q}} x_{qd(\nu)\nu} = 0, \ \nu \in \mathcal{V}. \tag{4.7}$$

Equation (4.2) ensures that each factory-visit node is used by at most one vessel. Equation (4.3) prescribes that each order be served by exactly one vessel. The constraints (4.4), (4.5), (4.6), (4.7) describe the routing flow and represent vessel starts at $o(\nu)$, their routing and end at $d(\nu)$. The vessels are required to pass through one of the factories before ending in the (possibly dummy) end-node $d(v)$, see equations (4.6), (4.7).

4.3. Vessel time constraints

$$x_{nn'\nu}(t_n + U_n a_{\nu n} + T_{nn'\nu} - t_{n'}) \leq 0, \ n \in \mathcal{N}_\nu^o, n' \in \mathcal{N}_\nu^d, \nu \in \mathcal{V}, \tag{4.8}$$

$$T_n^{min} \leq t_n \leq T_n^{max}, \ n \in \mathcal{N}, \tag{4.9}$$

$$t_{o(\nu)} = 0, \ \nu \in \mathcal{V}. \tag{4.10}$$

The non-linear time constraints (4.8) include travel and service times. The starting time of the service at node n cannot be less than the sum of the starting time and the service time at node n' and the sailing time from n to n' with ship ν, if ship ν is really sailing between those two nodes. The constraint can be linearised as:

$$t_n + U_n a_{\nu n} + T_{nn'\nu} - t_{n'} \leq (1 - x_{nn'\nu})T.$$

It is active only if order n' is served directly after order n. To make the formulation tighter, it is possible to decrease the upper bound T using time-window reduction algorithms, see [13]. Hard time windows, which cannot be violated, are given by (4.9). Finally, starting time is initialized in (4.10).

4.4. Cargo flow constraints

$$
\begin{aligned}
x_{nj\nu}(y_{\nu n} + a_{\nu j} - y_{\nu j}) &= 0,\ n \in \mathcal{N}^o_\nu, j \in \mathcal{J}, \nu \in \mathcal{V}, && (4.11)\\
x_{nq\nu}(y_{\nu n} - a_{\nu q} - y_{\nu q}) &= 0,\ n \in \mathcal{N}^o_\nu, q \in \mathcal{Q}_\nu, \nu \in \mathcal{V}, && (4.12)\\
a_{\nu o(\nu)} &= 0, \nu \in \mathcal{V}, && (4.13)\\
y_{\nu o(\nu)} &= 0, \nu \in \mathcal{V}, && (4.14)\\
y_{\nu j} &\leq K_\nu \sum_{n \in \mathcal{N}^o_\nu} x_{nj\nu},\ j \in \mathcal{J}, \nu \in \mathcal{V}, && (4.15)\\
a_{\nu j} &\leq y_{\nu j},\ j \in \mathcal{J}, \nu \in \mathcal{V}, && (4.16)\\
y_{\nu q} + a_{\nu q} &\leq K_\nu \sum_{n \in \mathcal{N}^d_\nu} x_{qn\nu},\ q \in \mathcal{Q}_\nu, \nu \in \mathcal{V}, && (4.17)\\
D_q &\leq \sum_{\nu \in \mathcal{V}} a_{\nu q},\ q \in \mathcal{Q}_\nu. && (4.18)
\end{aligned}
$$

Since the ships usually perform multiple trips during the time horizon, it is necessary to include ship inventory balance constraints explicitly. Equations (4.11) correspond to loading at a depot and (4.12) to unloading at a customer, if ship ν arrived to node j, or q respectively. The constraints can be linearised as:

$$
\begin{aligned}
y_{\nu n} + a_{\nu j} - y_{\nu j} &\leq (1 - x_{nj\nu})K_\nu,\\
y_{\nu n} + a_{\nu j} - y_{\nu j} &\geq -(1 - x_{nj\nu})K_\nu,\\
y_{\nu n} - a_{\nu q} - y_{\nu q} &\leq (1 - x_{nq\nu})K_\nu,\\
y_{\nu n} - a_{\nu q} - y_{\nu q} &\geq -(1 - x_{nq\nu})K_\nu.
\end{aligned}
$$

We assume that the ships are empty at the beginning, i.e., the initial load condition for each ship is given by (4.13), (4.14). Constraints (4.15) ensure that ship ν can be loaded only at a depot and the amount loaded must respect the ship capacity. Constraints (4.16) serve as valid inequalities, i.e., they can be omitted from the model, and impose a condition that the amount after loading is higher than the amount loaded. The sum of the unloaded amount and the amount left on the ship after that must not exceed the ship capacity by condition (4.17). Constraint (4.18) ensures that each order is satisfied; the combination with equation 4.3 guarantees that each order be covered by exactly one delivery, i.e., no split deliveries are admissible in the model.

5. Solutions methods

Solving the model directly with a MIP solver turned out to be possible for instances not larger than around 15 nodes. This can be understood given the number of binary variables and the NP-hardness of the problem. We have formulated the model in the GAMS modelling environment and used the CPLEX Solver.

In order to solve the real-sized EWOS case, we have resorted to heuristic approaches. We have gradually developed three heuristics – a construction heuristic, a heuristic based on tabu search and a clustering heuristic. They are described in Sections 5.1, 5.2 and 5.3 below. We have first developed the construction heuristic, with the aim to obtain solutions which are both feasible (if such solutions exist) and good. The quality, however, was not completely satisfactory and we could see that further improvements of the solution were necessary. We have implemented the tabu search heuristic and used the construction heuristic to create the initial solution. Eventually, we replaced the construction heuristic with the clustering heuristic, which was faster and performed better in combination with the tabu search.

5.1. Construction heuristic

The main idea is to construct routes for vessels by sequential insertion of orders to the end of open (not finished) routes. If the capacity of a ship is depleted, the closest depot is found and the ship can be refilled and restarted at this depot. The heuristic algorithm flow is described in Table 1 below. In the first step, all orders with a common time window are selected. After that, we try to find the best insertions to the end of the open routes. The first used criterion (marked by [1] in Table 1) is equal to the time distance between the customer with an unsatisfied order and the last node of the current open route. The second criterion (marked by [2] in Table 1) is based on the best and the second best possible insertions according to the first criterion. If no second insertion is possible for some nodes, then all nodes with second possible insertion are omitted in this step. Then, the node with the smallest distance is inserted. This criterion corresponds to some kind of regret when urgency plays an important role.

5.2. Tabu Search

Tabu search [17] is a meta heuristic originally proposed by Glover in 1986 [16] that has proved to be successful in tackling many hard optimization problems. It is an iterative neighbourhood search procedure which tries to escape local optima by temporarily accepting moves towards worse solutions and banning returning (improving) moves. The meta heuristic internally uses two main heuristic elements – an initial solution heuristic and a neighbourhood heuristic. The initial solution heuristic (e.g. construction, greedy, etc.) is used to find a starting point. The neighbourhood heuristic, usually given by a set of "moves", defines the neighbourhood of each point in the solution space.

Several tabu search heuristics have been proposed for the different variants of VRPs, see e.g. [1, 6, 9–11]. [6] gives a comprehensive overview of various tabu search heuristics for VRP with time windows together with their initial solution and neighbourhood heuristics. However, given the rich set of features which the VRP under our investigation exhibits (cf. Section 3), it was necessary for us to come up with a custom tabu search heuristic. The principles of our implementation are outlined below. We use the notation of Section 4 unless stated otherwise.

Every implementation of tabu search requires a data model of the solution, and an evaluation of the constraints and optimization criteria. The main data-concept in our model is a voyage, which is a combination of vessel (ν) and route (r). A route is a sequence of node (n) visits, $r = (n_1, n_2, \ldots, n_m)$. The firstand last nodes

Table 1. Construction heuristic.

0. Initialize open routes $\mathcal{O}$:
    Use starting depot for each ship
    and earliest time when it become available
  Unsatisfied orders $\mathcal{U} = \emptyset$
  Finished (closed) routes $\mathcal{C} = \emptyset$
  Sort orders $\mathcal{Q}$ w.r.t. time windows
  Sort ships $\mathcal{V}$ w.r.t. capacities

1. While $\mathcal{Q}$ is not empty do:
    Select all nodes with the same Time Windows: $\mathcal{Q}_{TW}$
    Set $\mathcal{Q} = \mathcal{Q} \setminus \mathcal{Q}_{TW}$

2. For each $q \in \mathcal{Q}_{TW}$ do:
      For each $o \in \mathcal{O}$ do:
        If insertion of q to the end of open route o is feasible
        with respect to time windows and ship capacity:
          If the insertion is the best with respect to the selected criterion[1]:
            Set $o(q) = o$ and the used ship $v(q) = v$
      End For
    End For

3. If no feasible insertion exists:
      Put the remaining orders into the set of unsatisfied customers: $\mathcal{U} = \mathcal{U} \cup \mathcal{Q}_{TW}$
      and set $\mathcal{Q}_{TW} = \emptyset$
    Else:
      Find the best feasible insertion $\hat{q}$ among all insertions $o(q)$ with respect to the
      selected criterion[2] and insert the order to the end of the route $o(\hat{q})$

4. If the capacity tolerance of ship $v(\hat{q})$ is reached after insertion:
        Finish the route: find the closest depot and insert it to the end
        Insert the finished route into the set of closed routes $\mathcal{C} = \mathcal{C} \cup \{o(\hat{q})\}$
        and remove it from open routes $\mathcal{O} = \mathcal{O} \setminus \{o(\hat{q})\}$
        Initialize new open route from the final depot with corresponding starting time

  End While

[1], [2] Two different criteria can be used.

are depots, not necessarily identical. Each of the intermediate nodes corresponds to one customer order and has the associated time-window $[T_n^{min}, T_n^{max}]$, which defines the requested earliest and latest arrivals.

Each node maintains four different time instants: the earliest arrival t_n^e, earliest departure o_n^e, latest arrival t_n^l and latest departure o_n^l. The earliest departure is obtained as earliest arrival plus unloading time

$$o_n^e = t_n^e + U_n a_n, \tag{5.1}$$

and symmetrically, the latest arrival is obtained as latest departure minus the unloading time

$$t_n^l = o_n^l + U_n a_n. \tag{5.2}$$

The parameter a_n denotes the delivered quantity (order size). It corresponds to the $a_{n\nu}$ parameter of the MIP model; the ν index was dropped here, since it was only necessary within the MIP formulation for modelling reasons. If nodes n and n' immediately follow each other, the earliest arrival to n' cannot be earlier than the earliest departure from n plus the necessary travel time

$$t^e_{n'} = \max\{T^{min}_{n'}, o^e_n + T_{nn'\nu}\}. \tag{5.3}$$

Symmetrically, the latest departure from node n cannot come later than the latest arrival to n' minus the necessary travel time

$$o^l_n = \min\{T^{max}_n + U_n a_n, e^l_{n'} - T_{nn'\nu}\}. \tag{5.4}$$

In this way, the earliest arrival and departure times are propagated from the first node forward, repeatedly using equations (5.1) and (5.3). The latest arrival and departure times are propagated from the last node backward, using equations (5.2) and (5.4). If the node sequence is such that $t^e_n \le t^l_n$ for all nodes n, then it is time-wise feasible. This time constraint is equivalent to the equations (4.8) and (4.9) in the MIP model.

The solution consists of a set of follow-up voyages for each vessel – when the vessel ends one voyage in a certain depot, it can either stop there completely, or load new cargo in this depot and start the next voyage from this depot. The routes corresponding to the follow-up voyages are linked for the forward and backward propagation of arrival and departure times (as explained above) to be performed along the entire node sequence from the first to the last route. The solution, feasible or infeasible, is always complete – i.e. every order-node is part of exactly one voyage (depot-nodes occur multiple times).

For notation purposes, we view the solution as a set of routes, where each route r has an associated vessel ν_r. We denote by L_r the total length of route r. The value of the solution is given by

$$\sum_r L_r + \beta_1 \sum_r \max\left\{0, \sum_{n\in r} a_n - K_{\nu_r}\right\} + \beta_2 \sum_r \sum_{n\in r} \max\left\{0, t^e_n - t^l_n\right\}.$$

The first term gives the total length of the routes in the solution. The second term is a penalty for violating the vessel capacity constraints, and the third term is a penalty for violating the time-constraints. The parameters $\beta_1 \ge 0$ and $\beta_2 \ge 0$ are penalty coefficients (see also below).

The neighbourhood of each solution is defined by the following moves:

Split voyage: Try to split every voyage on each position by each of the depots. That is, divide the voyage into two parts, make the vessel return to the chosen depot after the first part, and then let it start from the depot and finish the second part of the voyage. This move can be beneficial in restoring the feasibility when the voyage is currently infeasible due to the capacity constraint – when too many nodes are visited and too much cargo is carried.

Merge voyages: Try to combine each consecutive pair of follow-up voyages into one. That is, remove the visit to the depot between the two voyages and serve all the nodes of both voyages in one go.

Move node: For each node, take it out from its current position/voyage and try to insert it to all other positions in all voyages. Moves of this type make it possible to converge to voyages which are time-wise feasible (time windows are obeyed).

The flow of the tabu search algorithm is governed by tabu tenures and penalty coefficient adjustments. We use individual tabu tenures for each of the moves. The tabu tenure for split and merge of individual voyages is 20 iterations, the tabu tenure for moving each node is randomized, with the mean value equal to one third of the number of nodes. In each iteration, the penalties β_1 and β_2 are increased by 20% or decreased by 80% if the corresponding constraints are violated or satisfied, respectively.

As the initial heuristic, we at first used the construction heuristic described in Section 5.1, but eventually switched to the clustering heuristic, where nodes closest to given depot are initially distributed among voyages originating from that depot. For details on the heuristic see Section 5.3 below.

The neighbourhoods which we use are rather large (complexity $O(n^2)$). This is due to the fact that the typical number of nodes to be served in the EWOS case is about 100 – 200, which is a rather small number compared to large-scale VRP problems solved nowadays. The large neighbourhoods make our solutions barely sensitive to the initial solution (as also commented upon in [6]).

5.3. Clustering heuristic

The steps of the heuristic are as follows:

(i) For each vessel, we find out its starting depot and the first time it becomes available in this depot. Create an "empty" voyage for the given vessel, starting at a given time in a given depot and returning immediately back to the same depot. The starting depots encountered in this step form the set of *active depots*.

(ii) We partition all order-nodes into disjoint sets, one set for each active depot, based on the distance to the closest active depot.

(iii) For each depot, we gather all voyages created in (i) starting in this depot and order them by departure time. The number of such voyages is denoted by N_d.

(iv) For each depot, we take the cluster of order-nodes belonging to this depot and order them by earliest arrival. We split the linearly ordered set into N_d consecutive groups of an approximately equal size and then assign the first group to the first voyage from (iii), the second group to the second voyage, etc.

Such a solution might not be feasible, both the time and capacity constraints might be violated. It did however turn out to cooperate very well with the tabu search heuristic presented above.

6. Testing with historical data

The main aim of our work was to use our solution methods with real-world data and find/quantify improvements to the routing. The EWOS company has devoted a considerable effort to collecting their historical routing data. Based on the collected data, we reconstructed the entire voyage sequences of the EWOS fleet realized in Q2–Q4 2010. We carried out the following simulation.

We picked a particular time instant, let the ongoing (real recorded) voyages finish one-by-one, and let the tabu search heuristic overtake the control of the routing. The individual vessels became available at times and at depots corresponding to the ends of the real routes which were intercepted by the chosen time instant. As the set of orders, we chose all the unsatisfied orders within a 10 days horizon. Additionally, we added all the orders beyond the 10 days horizon which were part of any historically recorded route containing orders from the base order set. Finally, we compared the set of historically recorded routes to the routes proposed by the heuristic. The summary of the results for four different cases is given in Table 2. Improvements in travelled distance close to 30% were achieved. The solution times for the test cases were approximately 2 hours (2.53GHz Intel Core 2 Duo, 4GB RAM, MAC OS).

Table 2. Test case summary.

Start	Horizon [days]	Served orders	Optimized distance [km]	Historical distance [km]	Improvement
2010-10-14	10	118	8680.7	12923.3	32.8%
2010-10-24	10	124	10405.6	14595.3	28.7%
2010-11-06	10	139	9935.9	13753.4	27.8%
2010-11-15	10	113	9015.7	12679.5	28.9%

The fleet consists of 10 vessels. Apart from the vessel M/S RUBIN with a capacity of 350 tons, the vessel capacities range nearly uniformly from 600 to 1650 tons, with an average capacity of about 1200 tons. The traveling speeds range from 17 to 24 km/h. The sizes of the customer orders are units of tons up to about 200 tons, with few exceptions of 500 tons. The average order size is about 80 tons. The distances traveled between consecutive locations are tens to hundreds of kilometers.

For one of the test cases, we present a more detailed comparison of the real recorded historical routes and the proposed optimized routes in Tables 3 and 4. It is interesting to compare the achieved vessel fill rates, which are mostly above 90%, compared to the historical fill rates, which were significantly lower. The three factories (depots) Florø, Halsa and Bergneset are abbreviated as F, H and B. The Arrival and Departure columns use a compacted date format where the year 2010 and month 11 are omitted. For example, read "17 11:10" as "2010-11-17 11:10".

Table 3. Proposed optimized routes.

Vessel	From	To	Departure	Arrival	Fill-rate	Days	Dist. [km]	Visited
M/S ARTIC FJORD	F	F	17 11:10	24 11:12	0.98	8.1	1512.0	28
M/S ARTIC LADY	F	F	17 21:46	23 13:48	0.94	7.9	1522.6	16
M/S ARTIC SENIOR	F	F	19 20:00	23 02:14	0.55	5.7	76.1	4
M/S FEED BALSFJORD	H	H	15 23:14	22 16:17	0.99	7.3	1398.1	14
M/S FEED TROMSØ	B	B	16 08:19	22 05:53	0.90	6.9	1037.0	15
M/S HOLMEFJORD	F	F	17 10:22	21 19:34	0.84	3.7	521.3	9
M/S MIKAL WITH	H	H	16 14:04	24 00:16	1.00	6.5	2038.1	20
M/S RUBIN	B	B	18 18:13	20 20:02	0.91	5.5	535.6	3
M/S SAFIR	B	B	18 13:10	23 01:53	1.00	5.6	375.0	4

Table 4. Recorded historical routes.

Vessel	From	To	Departure	Arrival	Fill-rate	Days	Dist. [km]	Visited
M/S ARTIC FJORD	F	F	17 07:10	19 18:15	0.47	2.5	888.4	14
M/S ARTIC FJORD	F	F	20 21:10	22 16:11	0.77	1.8	667.4	12
M/S ARTIC FJORD	F	F	22 21:04	24 00:55	0.31	1.2	267.4	6
M/S ARTIC SENIOR	F	F	18 02:10	21 13:15	0.77	3.5	1045.0	5
M/S ARTIC SENIOR	F	F	22 08:15	24 02:20	0.74	1.8	616.5	12
M/S FEED BALSFJORD	H	H	15 15:48	18 15:38	0.63	3.0	848.1	7
M/S FEED BALSFJORD	H	H	18 19:47	21 08:16	0.45	2.5	795.1	6
M/S FEED TROMSØ	B	H	15 15:00	18 19:00	0.40	3.2	719.5	6
M/S FEED TROMSØ	H	B	19 07:55	22 15:41	1.16	3.3	1188.2	14
M/S HOLMEFJORD	F	F	18 04:52	20 12:10	0.17	2.3	492.1	4
M/S MIKAL WITH	H	H	17 19:55	21 09:30	0.36	3.6	1287.8	5
M/S MIKAL WITH	H	H	21 23:20	24 00:45	0.49	2.1	1422.7	10
M/S RUBIN	B	B	15 08:18	22 07:09	0.29	7.0	650.0	1
M/S SAFIR	B	B	17 17:00	19 23:00	0.55	2.2	614.9	6
M/S SAFIR	B	B	20 14:00	24 07:30	0.35	3.7	1176.3	5

One particular route listing is presented in table 5 below.

Table 5. M/S ARTIC FJORD, Fill-rate: 0.98, Days: 8.1, Distance: 1512.0 km.

Dist. [km]	Travel [h]	Wait [h]	Arrival	Departure	Deliv. [tons]	Destination
-	-	-	16 09:20	17 11:10	-	Florø
35.9	1.7	0.0	17 12:51	17 14:19	80.0	FUREVIKA
60.9	2.9	0.0	17 17:10	17 18:36	77.5	BÅRØYOSEN
135.2	6.3	0.0	18 00:56	18 01:34	34.0	HOLKAHOLO
158.6	7.4	0.0	18 09:01	18 09:28	25.0	SMALSKAR
7.4	0.3	0.0	18 09:49	18 11:30	92.5	NAUTVIK
6.0	0.3	0.0	18 11:47	18 12:09	20.0	KJERRINGA
162.8	7.6	0.0	18 19:48	18 20:54	59.6	HÅGARDSNESET
40.4	1.9	0.0	18 22:47	19 00:00	65.6	HÅVIK
23.2	1.1	8.2	19 09:19	19 10:03	40.0	ALDALEN
22.6	1.1	0.0	19 11:07	19 12:47	91.4	MIDTFLUA
0.0	0.0	0.0	19 12:47	19 12:47	0.0	MIDTFLUA
130.0	6.1	0.0	19 18:54	19 19:52	53.4	LEIHOLMANE
25.4	1.2	0.0	19 21:04	19 22:49	95.8	KJERRINGNESET
21.0	1.0	0.0	19 23:49	20 00:00	10.0	JUVIKA
51.3	2.4	17.7	20 20:07	20 20:32	22.0	EIDESBERGET
15.8	0.7	0.0	20 21:16	21 00:00	148.6	MJØLSVIK
129.4	6.1	4.4	21 10:26	21 12:05	90.0	OSPENESET, AUSTFJ.
5.0	0.2	0.0	21 12:19	21 13:19	55.0	REKEVIKI, AUSTFJ.
137.4	6.5	0.0	21 19:46	21 22:07	127.8	DJUPEVIKA
2.3	0.1	0.0	21 22:14	22 00:00	96.3	TOBBHOLMANE
36.1	1.7	38.9	23 16:32	23 16:54	20.0	ALDALEN
15.1	0.7	0.0	23 17:37	23 18:10	30.0	NYGÅRD
21.1	1.0	0.0	23 19:09	23 19:42	30.0	DJUPEDALEN
8.9	0.4	0.0	23 20:07	23 20:40	30.0	SKÅTAVÅGEN
16.9	0.8	0.0	23 21:28	23 22:01	30.0	MJÅNES
42.1	2.0	0.0	24 00:00	24 00:55	50.0	USHOLMSVIKA
19.8	0.9	0.0	24 01:50	24 01:57	6.0	FIKSNESET
93.4	4.4	0.0	24 06:20	24 07:04	40.0	BÅRØYOSEN
88.1	4.1	0.0	24 11:12	-	-	Florø

The algorithm seemed to find a very good overall solution in terms of the total traveled distance and the fill rate. It uses fewer routes which are longer, compared to the reality where a bigger number of shorter routes were used. The algorithm may propose a route with a waiting time in the middle (e.g. the 38.9 hour waiting time in the route above). This could be counter-intuitive for a human planner, but in this case it still results in a more effective coverage of the sequence of customer requests. The waiting times are of course a consequence of the time windows on the customer orders.

7. Conclusions and further research

We have presented a MIP model, a construction heuristic, a clustering heuristic and a tabu search heuristic for a rich vehicle routing problem, which arises in distribution planning of the Norwegian company EWOS AS; a fish feed producer for the salmon farming industry. The size of the problem made it impossible to use exact solution methods, it mde it possible, however, to implement a tabu search heuristic with a rather large neighbourhood search. The obtained results are satisfactory from the practical viewpoint – the solution resulted in a significant reduction of the travelled distance (close to 30%) and an increase of average vessel fill-rate (from 60% up to 95%), as compared to the real company data. The results also indicate a potential for down-scaling the fleet, with additional considerable cost savings for the company.

References

[1] C. Archetti, A. Hertz and M. G. Speranza, *A tabu search algorithm for the split delivery vehicle routing problem*, Transport. Sci. **40** (2006), 64–73.

[2] N. Azi, M. Gendreau and J.-Y. Potvin, *An adaptive large neighborhood search for a vehicle routing problem with multiple routes*, Comput. Oper. Res. **41** (2014), 167–173.

[3] R. Baldacci, M. Battarra and D. Vigo, *Routing a heterogeneous fleet of vehicles*, in: B. L. Golden, S. Raghavan and E. A. Wasil (eds.), The Vehicle Routing Problem: Latest Advances and New Challenges, Operations Research/Computer Science Interfaces **43**, Springer, 2008, 3–27.

[4] M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser (eds.), *Network Routing*, Handbooks in Operations Research and Management Science **8**, Elsevier, 1995.

[5] C. Barnhart and G. Laporte (eds.), *Transportation*, Handbooks in Operations Research and Management Science **14**, Elsevier, 2007.

[6] O. Bräysy and M. Gendreau, *Tabu search heuristics for the vehicle routing problem with time windows*, TOP: An Official Journal of the Spanish Society of Statistics and Operations Research **10** (2002), 211–237.

[7] D. Cattaruzza, N. Absi, D. Feillet and T. Vidal, *A memetic algorithm for the multi trip vehicle routing problem*, Eur. J. Oper. Res. **236** (2014), 833–848".

[8] M. Christiansen, K. Fagerholt, B. Nygreen and D. Ronen, *Maritime transportation*, in: C. Barnhart and G. Laporte (eds.), Handbooks in Operations Research and Management Science: Transportation **14**, Elsevier, 2007, 189–284.

[9] J.-F. Cordeau, M. Gendreau and G. Laporte, *A tabu search heuristic for periodic and multi-depot vehicle routing problems*, Networks **30** (1997), 105–119.

[10] J.-F. Cordeau and G. Laporte, *A unified tabu search heuristic for vehicle routing problems with time windows*, J. Oper. Res. Soc. **52** (2001), 928–936.

[11] J.-F. Cordeau and G. Laporte, *Tabu search heuristics for the vehicle routing problem*, in: R. Sharda, S. Voss, C. Rego and B. Alidaee (eds.), Metaheuristic Optimization via Memory and Evolution Tabu Search and Scatter Search, Springer, 2005, 145–163.

[12] G. B. Dantzig and J. H. Ramser, *The truck dispatching problem*, Manag. Sci. **6** (1959), 80–91.

[13] J. Desrosiers, Y. Dumas, M. M. Solomon and F. Soumis, *Time constrained routing and scheduling*, in: M. Ball, T. Magnanti, C. Monma and G. Nemhauser (eds.), Handbooks in Operations Research and Management Science: Network Routing **8**, Elsevier, 1995, 35–139.

[14] R. Dondo and J. Cerd, *A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows*, Eur. J. Oper. Res. **176** (2007), 1478–1507.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.

[16] F. Glover, *Future paths for integer programming and links to artificial intelligence*, Comp. Oper. Res. **13** (1986), 533–549.

[17] F. Glover, *Tabu search: A tutorial*, Interfaces **20** (1990), 74–94.

[18] S. C. Graves, A. H. G. Rinnooy Kan and P. H. Zipkin (eds.), *Logistics of Production and Inventory*, Handbooks in Operations Research & Management Science **4**, Elsevier, 1993.

[19] G. Gutin and A. P. Punnen (eds.), *The traveling salesman problem and its variations*, Combinatorial Optimization **12**, Kluwer Academic Publishers, 2004.
[20] A. Hoff, H. Andersson, M. Christiansen, G. Hasle and A. Løkketangen, *Industrial aspects and literature survey: Fleet composition and routing*, Comp. Oper. Res. **37** (2010), 2041–2061.
[21] S. Lin, *Computer solutions of the traveling salesman problem*, Bell Syst. Tech. J. **44** (1965), 2245–2269.
[22] R. Macedo, C. Alves, J. M. V. de Carvalho, F. Clautiaux and S. Hanafi, *Solving the vehicle routing problem with time windows and multiple routes exactly using a pseudo-polynomial model*, Eur. J. Oper. Res. **214** (2011), 536–545.
[23] J. Mulder and R. Dekker, *Methods for strategic liner shipping network design*, Eur. J. Oper. Res. **235** (2014), 367–377.
[24] G. Pantuso, K. Fagerholt and L. M. Hvattum, *A survey on maritime fleet size and mix problems*, Eur. J. Oper. Res. **235** (2014), 341–349.
[25] W. B. Powell and H. Topaloglu, *Stochastic programming in transportation and logistics*, in: A. Ruszczynski and A. Shapiro (eds.), Handbooks in Operations Research and Management Science: Stochastic Programming **10**, Elsevier, 2003, 555–635.
[26] S. Salhi, A. Imran and N. A. Wassan, *The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation*, Comp. Oper. Res. **52** (2014), 315–325.
[27] L. Santos, J. Coutinho-Rodrigues and C. H. Antunes, *A web spatial decision support system for vehicle routing using Google maps*, Decis. Support Syst. **51** (2011), 1–9.
[28] M. Savelsbergh and J.-H. Song, *Inventory routing with continuous moves*, Comp. Oper. Res. **34** (2007), 1744–1763.
[29] M. Savelsbergh and J.-H. Song, *An optimization algorithm for the inventory routing problem with continuous moves*, Comp. Oper. Res. **35** (2008), 2266–2282.
[30] M. M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Oper. Res. **35** (1987), 254–265.
[31] P. Toth and D. Vigo (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications **9**, SIAM Society for Industrial and Applied Mathematics, Philadelphia, 2002.
[32] T. Vidal, T. G. Crainic, M. Gendreau and C. Prins, *Implicit depot assignments and rotations in vehicle routing heuristics*, Eur. J. Oper. Res. **237** (2014), 15–28.
[33] L. A. Wolsey, *Integer Programming*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, 1998.

Martin Branda, Department of Probability and Mathematical Statistics, Faculty of Mathematics and Physics, Charles University, Sokolovská 83, 186 75 Prague, Czech Republic
e-mail: `branda@karlin.mff.cuni.cz`

Kjetil K. Haugen, Faculty of Logistics, Molde University College, PO.Box 2110, 6402 Molde, Norway
e-mail: `kjetil.haugen@himolde.no`

Jan Novotný, Institute of Mathematics, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2, 616 69 Brno, Czech Republic
e-mail: `iannovot@gmail.com`

Asmund Olstad, Faculty of Logistics, Molde University College, PO.Box 2110, 6402 Molde, Norway
e-mail: `asmund.olstad@himolde.no`