Centro de Investigação Operacional

Routing in fishery research: a metaheuristic approach

Marta Mesquita, Alberto G. Murta, Ana Paias and Laura Wise

CIO – Working Paper 3/2015

Routing in fishery research: a metaheuristic approach

Marta Mesquita[1,4], Alberto G. Murta[*], Ana Paias[2,4] and Laura Wise [3]

[1] ISA, ULisboa, Tapada da Ajuda,
1349-017 Lisboa, Portugal. E-mail: martaoliv@isa.ulisboa.pt

[2] DEIO, Faculdade de Ciências, ULisboa, C6, Piso 4,
1749-016 Lisboa, Portugal. E-mail: ampaias@fc.ul.pt

[3] Instituto Português do Mar e da Atmosfera, Av. Brasília,
1449-006 Lisboa, Portugal. E-mail: lwise@ipma.pt

[4] Centro de Investigação Operacional, Faculdade de Ciências, ULisboa

March 4, 2015

Abstract

Every autumn, a research vessel carries out a sampling survey trip to estimate the abundance of groundfish species of the Portuguese continental waters. The sampling operations are made at predefined geographical locations, the fishing stations, within predefined multiple time windows. The vessel route starts and ends at the port of Lisbon and must visit all fishing stations. According to a predefined periodicity, the vessel must enter a port to supply food, refuel and/or change crew. Given the geographical locations of the fishing stations/ports and the current weather conditions, the objective is to minimize the total traveled distance and the completion time. We present a Mixed Integer Linear Program to describe the problem and propose three heuristic approaches, that combine genetic algorithms and Adaptive Large Neighborhood Search, to solve it. Computational experience with real data shows that the proposed heuristics are suitable tools to solve the problem.

Keywords: Traveling salesman problem, multiple time windows, genetic algorithm, adaptive large neighborhood search, heuristics.

[*]In memory of Alberto G. Murta.

1 Introduction

Since 1979, IPMA carries out bottom trawl surveys that cover the Portuguese continental coast from Caminha to Vila Real de Santo Antonio. Primarily, these surveys are used as fisheries independent tools to estimate abundances, recruitment and geographical distribution of several demersal fish species. These surveys help to assess the status of important commercial species such as hake (*Merluccius merluccius*) and horse mackerel (*Trachurus trachurus*) and also to monitor other fish and crustacean species such as blue whiting (*Micromesistius poutassou*), mackerel (*Scomber scombrus*), Norway lobster (*Nephrops norvegicus*), deepwater rose shrimp (*Parapenaeus longirostris*), among others. Surveys are carried out once a year in Autumn and last around 28 days. Sampling is made using a Norwegian Campelen Trawl and sample size is around 95 locations. Sample sites are fixed from year to year along the Portuguese continental shelf and slope. Normally, in each sample site location (fishing station) only one haul is carried out: the trawl is set on the bottom of the sea, towed for 30 minutes and hauled back into the ship. The duration of the haul lasts around 60 minutes, depending on each fishing station depth because at greater depths more time is needed to set and retrieve the trawl. The catch (species caught) is then sampled on board: each species is weighted and counted separately. Target species are always measured and weighed by length class. Whenever there is time available all other species are also measured. When catches are very high, a representative sub-sample is deemed necessary. Other individual biological parameters are also collected for the target species: maturity, sex-ratio, food habits, etc. Catch sampling is carried out while the vessel moves from one fishing station to the next fishing station.

According to their geographical locations, north, southwest and south coast, the set of fishing stations is partitioned into two subsets. Each subset gives rise to a survey circuit, starting and ending at the port of Lisbon, which should be completed in 14 or less days. Depending on the current weather conditions, the captain decides whether to start sampling the northern half of the coast, or the southern one. The other half is sampled in the second part of the survey. In each part of the survey, based on the captains experience, the sequence of fishing stations is determined on board depending on factors such as sea state, wind, time of departure, etc. Usually, when the vessel leaves the port it heads to the nearest fishing station. After finishing the fishing operation, the vessel proceeds to the next station. All fishing operations must be carried out during daytime (from 7 am to 6 pm). Consequently, during the night the vessel just navigates from the last fishing station of the day to the first fishing station of the next day. This routine goes on for 7 days, after which the vessel must enter a port. The vessel has authorization to stop at Portimão (Portim) in the south, Lisbon (Lis) in the center and Figueira da Foz (Fig) or Matosinhos (Matos) in the north. The vessel must arrive at one of these ports by 6 pm and leaves port the next morning, usually before dawn, in order to arrive at the first fishing station just before sunrise. After finishing the first part of the survey, the vessel returns to the port of Lisbon for food and water supply and, eventually, to change crew.

The second part of the survey is similar to the first one, covering the fishing stations that remain to be sampled.

Environmental concerns call for the minimization of traveled distance. Actually, minimizing the traveled distance results in less fuel and oil consumption which in turn reduces environmental emissions as well as sea pollution. Due to time windows constraints associated with the fishing operations, the shortest path in distance will not necessarily be the shortest path in time. From the management and staff point of view it is desirable to minimize the completion time in order to return home as soon as possible. To achieve the above purposes a mathematical model, describing the underlying problem, will be developed and optimization techniques will be used to deal with it.

Given the geographical locations of the fishing stations and of the ports (see Figure 1), and the average speed of the vessel, the ship route optimization problem (SROP) consists of determining two circuits covering the whole sampling plan that minimize the total traveled distance as well as the completion time of each circuit such that:

- the first circuit starts on day 1 and finishes before the end of the 14th day, at the port of Lisbon;
- the second circuit starts on day 15 and finishes before the end of the 28th day, at the port of Lisbon;
- each fishing station is visited exactly once by one of the circuits;
- at each fishing station the fishing operation starts between 7 am and 6 pm; If the vessel arrives at a fishing station before 7 am then it has to wait until 7 am to start a fishing operation.
- on the 7th and 21st days, the vessel enters a port (Lis, Portim, Fig or Matos) where it, eventually, fills with fuel and stocks up with food and water. It leaves port the following day;

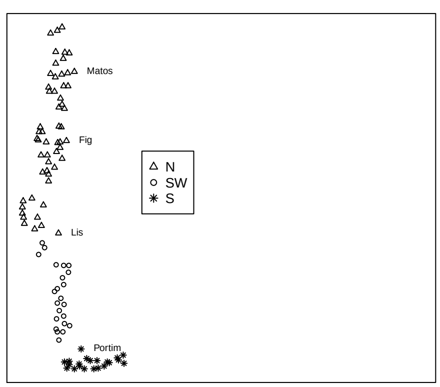

Figure 1: Fishing stations and ports

The SROP may be viewed as a variant of the Traveling Salesman Problem (TSP) with multiple time windows. In fact, there are several well known variants of the TSP in which not all clients need to be visited (see [2] for a survey on TSP variants). In the Generalized TSP cities are partitioned into clusters and the objective is to find the minimum cost circuit visiting at least one city of each cluster. If we consider that each fishing station defines a cluster and all the ports are included in an additional cluster then the SROP can be seen as variant of the Generalized TSP with time windows and additional constraints where each 'mandatory' cluster is visited once and the 'selective' cluster is visited more than once according to a predefined periodicity. The SROP also has some similarities with extensions of the Orienteering Problem in which each node has associated a non-negative weight. The objective of the Orienteering Problem is to find a path, whose total traveled time does not exceed a given threshold value, visiting the subset of cities that maximizes the total weight. If the goal is to find P paths (circuits), each limited in time by a predefined threshold value, the problem is known as the Team Orienteering Problem. In [4] the authors proved that the Orienteering Problem is NP-hard and developed a center-of-gravity heuristic. Many other researchers propose heuristics to tackle this problem - see [13] for a survey on the Orienteering Problem. In particular, heuristic algorithms to solve the (Team) Orienteering Problem with time windows have recently been proposed by [12, 11, 6] and [5]. In particular, [5] proposes a genetic algorithm for solving the Orienteering Problem with time windows while [11] presents a Variable Neighborhood Search procedure to solve a multi-period Orienteering Problem with multiple time windows where mandatory clients should be visited exactly once and potential customers can be visited at most once. In [8] the authors generalize the Orienteering Problem by allowing node rewards and arc length to vary based on the amount of the resources expended at each node and propose a branch and bound algorithm to solve the problem.

When the starting and final locations of the path are the same, the Orienteering Problem looks for a circuit and is often called Selective TSP. In [3] the authors developed different classes of valid inequalities and used a branch-and-cut algorithm to solve a Selective TSP that includes a subset of compulsory cities.

Furthermore, the SROP may be viewed as a real world application of the TSP presented in [7], namely the TSP with selective cities and multiple time windows (TSPSTW). In the TSPSTW, the set of cities is partitioned into two subsets: mandatory cities and selective cities. All mandatory cities should be visited once within one of the corresponding predefined multiple time windows. A subset of the selective cities, whose cardinality depends on the tour completion time, should be visited within one of the associated multiple time windows. The objective is to plan a tour, not exceeding a predefined number of days, that minimizes a linear combination of the total traveled distance as well as the completion time. For the current application the mandatory cities correspond to the fishing stations and the selective cities correspond to ports.

The outline of this paper is as follows. In section 2 the SROP is formally stated and a mathematical formulation is proposed. Section 3 is devoted to the solution methodology. Computational results for different scenarios are reported and discussed in section 4. Finally, some conclusions are drawn in section 5.

2 Problem formulation

To study the abundance of fish species, a research vessel must visit a set of fishing stations. The survey trip is partitioned into two circuits, starting and ending at the port of Lisbon, and lasting at most 14 days. According to a predefined periodicity of 7 days, each circuit must visit one of the p available ports to supply with food, refuel and/or change crew.

Therefore, one wants to determine two circuits that cover exactly once each fishing station within predefined time windows:

$$\begin{array}{c}\text{Lisbon} \longrightarrow \text{fish station set 1} \longrightarrow \text{port 1} \longrightarrow \text{fish station set 2} \longrightarrow \\ \text{Lisbon} \longrightarrow \text{fish station set 3} \longrightarrow \text{port 2} \longrightarrow \text{fish station set 4} \longrightarrow \text{Lisbon}\end{array}$$

More formally, given n fishing stations and p ports, consider graph $G = (V, A)$. The set $V = E \cup P \cup L$ is the vertex set in which each vertex $i \in E = \{1, ..., n\}$ corresponds to a fishing station and each vertex $i \in P = \{n + 1, ..., n + p\}$ corresponds to a port. Set $L = L_1 \cup L_2$, with $L_1 = \{0, n + p + 1\}$ and $L_2 = \{n + p + 2, n + p + 3\}$, includes replicas of the port of Lisbon to represent the starting and ending location of the route in the first circuit, L_1, and the starting and ending location of the route in the second circuit, L_2. We associate a commodity with each circuit. For each commodity $k = 1, 2$ there is an arc set A^k that includes arcs linking any pair of fishing stations, $(\{(i, j) : (i, j) \in E \times E, i \neq j\}$, arcs connecting a fishing station to a port and vice versa, $\{(i, j) : (i, j) \in E \times P\}$ and $\{(i, j) : (i, j) \in P \times E\}$, and arcs linking a fishing station to vertices corresponding to Lisbon, $\{(i, j) : i \in L_1, j \in E\}$ and $\{(i, j) : i \in E, j \in L_2\}$. The arc set $A = A^1 \cup A^2$.

A travel time $t_{i,j}$, indicating the time spent traveling from location i to location j, is associated to each arc $(i, j) \in A$. Note that travel times need not satisfy the triangle inequality and need not be symmetric.

For each day $h = 1, ..., 14$ a time window $[e^h, l^h]$ is defined to ensure that, in each circuit, for each fishing station visited on day h, the fishing operation starts within 7 am and 6 pm and occurs before the end of the 14th day. In particular, for day 1, 2, ..., 14 corresponds, respectively, the time window $[7, 18]$, $[31, 42]$, ..., $[319, 330]$, established in hours. Each vertex $i \in E$ has to be visited within exactly one of the 14 time windows. Moreover, time windows $[e^{14+s}, l^{14+s}]$, $s = 1, 2$ are defined for the possible visits to a port in the first and/or the second circuit. In this application, a predefined periodicity $d = 7 \times 24$ (hours) sets that after 7 consecutive days sailing a port should be visited. For each vertex $i \in E \cup P$ the time spent during the corresponding visit is given and denoted by $prof_i$.

To model the SROP we propose a mathematical formulation that is tailor-made for sequential solutions approaches. We consider three types of decision variables: clustering variables, route variables and time variables. Concerning clustering variables let $y_i^k = 1$ if fishing station i is sampled during circuit k and 0 otherwise. As for the route variables, let $x_{i,j}^k = 1$ if the vessel travels directly from vertex i to vertex j in circuit k and 0 otherwise. Two sets of decision variables are used to represent the time variables. One set defines the time window at which the visit to each fishing station or to a port occurs. That is, $\delta_i^h = 1$ if fishing station $i \in E$ is visited during time window $h, h = 1, ..., 14$ and 0 otherwise. Regarding the visit to a port at the end of the first and third week, let $\gamma_i^k = 1$ if port i is visited during circuit (time window) $k = 1, 2$ and 0 otherwise. Binary variables z^k, $k = 1, 2$, that depend on the predefined periodicity d and the duration of circuit k, define if a mandatory visit to a port in circuit k should be made or not. The other set of variables includes continuous time decision variables, w_i, $i \in E \cup P$, to indicate the starting time of the operation in vertex i. Moreover, variables w_0 and w_{n+p+2} define the starting time for the first and second circuit while w_{n+p+1} and w_{n+p+3} define the arrival time to Lisbon at the end of the first and second circuit. Parameter $b = 14 \times 24$ (hours) defines the latest arrival time at Lisbon for each circuit .

The SROP can be described by the following Mixed Integer Linear Programming (MILP) formulation:

$$min \sum_{(i,j)\in A} t_{ij}\ x_{ij}^{\ell} + (w_{n+p+3} - w_{n+p+2}) + (w_{n+p+1} - w_0) \tag{1}$$

$$\sum_{k=1}^{2} y_i^k = 1 \quad i \in E \cup L \tag{2}$$

$$\sum_{i:(i,j)\in A^k} x_{ij}^k = y_j^k \quad j \in E \cup \{n+p+1, n+p+3\}, k = 1,2 \tag{3}$$

$$\sum_{j:(i,j)\in A^k} x_{ij}^k = y_i^k \quad i \in E \cup \{0, n+p+2\}, k = 1,2 \tag{4}$$

$$\sum_{j:(i,j)\in A^k} x_{ij}^k - \sum_{j:(j,i)\in A^k} x_{ji}^k = 0 \quad i \in P, k = 1,2 \tag{5}$$

$$\sum_{h=1}^{14} \delta_i^h = 1 \quad i \in E \tag{6}$$

$$\sum_{h=1}^{14} e^h \delta_i^h \le w_i \le \sum_{h=1}^{14} l^h \delta_i^h \quad i \in E \tag{7}$$

$$z^1 \ge \frac{w_{n+p+1} - w_0}{d} - 1 \tag{8}$$

$$z^2 \ge \frac{w_{n+p+3} - w_{n+p+2}}{d} - 1 \tag{9}$$

$$\sum_{j\in P} \gamma_j^k = z^k \quad k = 1,2 \tag{10}$$

$$\sum_{k=1}^{2} e^{14+k} \gamma_j^k \le w_j \le \sum_{k=1}^{2} l^{14+k} \gamma_j^k \quad j \in P \tag{11}$$

$$\gamma_j^k = \sum_{i:(i,j)\in A^k} x_{i,j}^k \quad j \in P, k = 1,2 \tag{12}$$

$$w_j = 0 \quad j = 0, n+p+2 \tag{13}$$

$$w_j \le b \quad j = n+p+1, n+p+3 \tag{14}$$

$$w_j \ge w_i + prof_i + \bar{t}_{ij} - M(1 - x_{ij}^k) \quad (i,j) \in A^k, k = 1,2 \tag{15}$$

$$x_{ij}^k \in \{0,1\} \quad (i,j) \in A^k, k = 1,2 \tag{16}$$

$$w_i \ge 0, \quad i \in E \cup P \tag{17}$$

$$\delta_i^h \in \{0,1\} \quad i \in E, h = 1, ..., 14 \tag{18}$$

$$\gamma_i^k \in \{0,1\} \quad i \in P, k = 1,2 \tag{19}$$

$$y_i^k \in \{0,1\} \quad i \in E \cup L, k = 1,2 \tag{20}$$

$$z^k \ge 0 \quad integer, \ k = 1,2 \tag{21}$$

The objective function (1) involves the minimization of two terms: the total traveled time and the completion time, which correspond respectively to save fuel and oil consumption and to save time. Note that the duration of each circuit, $(w_{n+p+3} - w_{n+p+2})$ and $w_{n+p+1} - w_0$, is a function of the vessel's travel time, the time waiting at each station before starting the fishing operation and the time spent at each port. As variables w_0 and w_{n+p+2} are set equal to zero, the duration of each circuit is equal to its completion time. Constraints (2), (3) and (4) state that the vessel visits once each fishing station and this visit occurs in a circuit that starts and ends at Lisbon. Moreover, (3) and (4) ensure flow conservation for vertices corresponding to fishing stations. Note that, we set a priori $y_0^1 = y_{n+p+1}^1 = y_{n+p+2}^2 = y_{n+p+3}^2 = 1$. Constraints (5) ensure flow conservation for vertices corresponding to ports. Constraints (6) ensure that each fishing station is visited within exactly one time window while (7) guarantee that the visit occurs within a feasible time window. Constraints (8), (9) and (10) define, for each circuit, the number of mandatory visits to the ports according to a given periodicity d. Constraints (11) guarantee that when a visit to a port occurs it must be within predefined time windows. Constraints (12) guarantee the consistency between variables $x_{i,j}^k$ and variables γ_j^k. Constraints (13) establish the time at which the vessel leaves Lisbon for the first and second circuits. Constraints (14) ensure that the duration of each circuit j is equal to or less than b. Since route variables are indexed by each circuit, we consider that each circuit starts at instant 0 and finishes before instant $b = 14 \times 24$ (hours). Constraints (15) link variables x and w. These constraints establish the precedence relation between two consecutive vertices visited by the vessel and eliminate sub-tours. In detail, constraints (15) state that if the vessel goes from location i to location j then the time at which it starts visiting j, w_j, must be greater than the time at which it starts visiting i, w_i, plus the time spent at vertex i, plus a function of the time spent in transit from i to j, $\bar{t}_{ij}$ whose expression is given by $\bar{t}_{ij} = t_{ij}$ if $i \in P$ or $\bar{t}_{ij} = \epsilon + t_{ij}$ if $i \in E$. Parameter ϵ accounts for extra time due to possible setbacks at a fishing station.

3 Solution approach

We propose three alternative meta-heuristics, based on genetic algorithms and Adaptive Large Neighborhood Search (ALNS), to solve the SROP.
In the mathematical model (1)-(21), presented in the previous section, one can identify three combinatorial sub-problems: a clustering problem dividing the set of fishing stations into two subsets; a routing problem defining the spatial movement of the vessel; a scheduling problem establishing the times at which each location is visited. Despites the existence of a high dependency among the referred three sub-problems one may devise a hierarchic ordering of the decisions that are involved in the resolution of the SROP which suggests the use of sequential approaches. We present two sequential approaches that divide the decisions involved in the problem resolution into two phases and an integrated approach that exploits the dependency of the referred three sub-problems. All approaches produce solutions that can be further optimized by using ALNS.

3.1 Sequential approaches

To solve the SROP, IPMA follows a sequential approach based on a hierarchic ordering of decisions. At the beginning of the survey trip, a clustering decision has to be made to define the partition of the set of fishing stations into two subsets E_1 and E_2. Fishing stations belonging to E_1 are visited during the first two weeks while fishing stations in E_2 are surveyed during the third and fourth week. For each subset E_i, $i = 1, 2$, a routing decision, states the sequence of fishing stations to be sampled. After that, a scheduling decision establishes the times at which each location is visited on each day. Note that concerning IPMA's approach, decisions of one level are taken assuming that the previous levels decisions have already been made and will constraint decisions of the subsequent levels. In subsection 3.1.1, we propose two new alternative sequential approaches, Seq and SeqPlus that use the same hierarchic ordering of the decisions but allow different sub-problem integration between two levels of decision. In 3.1.2 we refer a sequential approach already presented in [7] which will be compared with the new proposals from a computational point of view.

3.1.1 Sequential approaches based on metaheuristics

In this subsection, we propose two alternative sequential approaches, Seq and SeqPlus both based on genetic algorithms. Table 1 summarizes for each approach and for each phase the underlying optimization sub-problem(s) to be solved and type of decisions considered.

	Approach Seq	**Approach SeqPlus**
Phase 1:	Solve clustering-routing Fixe clusters and route	Solve clustering-routing Fixe clusters
Phase 2:	Solve scheduling	Solve routing-scheduling

Table 1: Decision scheme

In Seq the clustering and routing decisions are fixed by phase 1 and phase 2 just deals with the scheduling subproblem. In SeqPlus only clustering decisions are fixed in phase 1 and phase 2 deals with the integrated routing and scheduling subproblem.

Concerning both sequential approaches, the goal of phase 1 is to make a partition of the set of fishing stations into two subsets taking into account the spatial movements of the vessel and minimizing the total traveled distance (that is the first term of the objective function 1). This is accomplish by determining two disjoint circuits sharing once the port of Lisbon and ensuring that each fishing station is covered once by one of the circuits, disregarding time windows constraints. Each circuit $i, i = 1, 2$ defines the subset of stations E_i. Each subset E_i, $i = 1, 2$, includes the fishing stations visited by circuit i.

In Seq the circuits obtained in phase 1 define both the two subsets E_1 and E_2 and the ordering by which the fishing stations are visited, disregarding the ports. Then, in phase 2, a greedy algorithm defines the starting time for the visit to each fishing station and establishes the visits to the ports, satisfying time windows.

In SeqPlus the circuits obtained in phase 1 just define the two subsets E_1 and E_2. Then, for each E_i, $i = 1, 2$, phase 2 solves an integrated routing and scheduling problem to optimize the traveled distance and the completion time of each circuit, while ensuring that time windows as well as the compulsory visits to ports are verified.

Next, we detail the procedures developed to implement Seq and SeqPlus. In phase 1 of both approaches, a genetic algorithm, further denoted by GenClusR, is used to determine the two disjoint circuits sharing once the port of Lisbon. Each solution is represented as a permutation of the set vertices $0, 1, ..., n, n + p + 1$, where the port of Lisbon is split into two vertices $0, n + p + 1$ which will indicate where the first circuit starts and ends. In general, the permutation,

$$i_1, ..., i_a, 0, j_1 ..., j_b, n + p + 1, k_1, ..., k_c$$

leads to the following partitioning of vertices into two circuits

$$0, j_1 ..., j_b, n + p + 1$$

and

$$n + p + 2, k_1, ..., k_c, i_1, ..., i_a, n + p + 3.$$

The fitness function, to be maximized, is defined as the reciprocal of the total tour length. An initial random (uniform) population is generated. Concerning the genetic operators, we have considered linear-rank selection, simple inversion mutation and cycle crossover (CyCx) operators. The pseudo-code of the algorithm GenClusR is presented in Algorithm 1 where the required parameters M, pc, pm, new%, maxgen represent,repectively, the population size, the crossover probability, mutation probability, the % of population to be replaced in each generation and the maximum number of generations.

Algorithm 1 GenClusR pseudo-code

```
Require: n, p, M, pc, pm, new%, maxgen
Generate an initial population of M random (n+2)-permutations.
Evaluate the population considering the reciprocal of the total tour length.
repeat
   Select parents from the population using linear-rank selection.
   Mate the parents to produce children: to each pair of parents apply, with
   probability pc, the crossover operator CyCx and apply to each children,
   with probability pm, the simple inversion mutator.
   Evaluate fitness of the children using the decoder and linear scaling.
   Substitute at most new% of the population by the children.
until maxgen is reached or population has converged
The best solution found is the one corresponding to the highest fitness.
```

In approach Seq, the two circuits of phase 1, define the order by which the fishing stations will be visited. Then it only remains to define the starting time associated to each visit and also the visits to the ports. So, starting in Lisbon at instant 0, the vessel moves towards the first station to be visited, if it arrives before the starting time of the corresponding time window then it has to wait until that instant to start the visit. After finishing the fishing operation at this station, the vessel moves toward the next station and, again, the validity of the time window constraint should be checked. The procedure goes on until the end of the seventh day is reached. Then, a compulsory visit to the nearest port is assured. After leaving the port the vessel moves to the next fishing station and the procedure continues until the end of the circuit, when the port of Lisbon is reached. The procedure does not guarantee that at the end of each circuit, the vessel returns to Lisbon before the 14th day. However, we anticipate that the low fitness score of solutions exceeding 14 days will rule them out from the population in earlier iterations of the genetic algorithm.

Procedure 2 Seq

1: Run GenClusR to obtain two disjoint circuits covering all the fishing stations
2: Considering the ordering of stations defined by the circuits in 1, obtain a solution to the SROP (Algorithm 4 Decoder).

In spite of phase 1 being similar for approaches SeqPlus and Seq, concerning SeqPlus the input for phase 2 is the set of of two clusters of fishing stations, E_i, $i = 1, 2$, induced by each circuit i obtained in phase 1. Consequently, in phase 2 we have to solve two partial SROPs each corresponding to a routing and scheduling subproblem for each cluster of stations. We solve each subproblem using the genetic algorithm, GenRSched described in Algorithm 3. GenRSched uses a permutation of the fishing stations to code each solution and a decoder, detailed in Algorithm 4, to assess the corresponding value of the objective function (1). The decoder is similar to the heuristic procedure described in phase 2 of the approach Seq. Given a permutation of the fishing stations, the decoder determines the starting time of each visit, assuring the validity of time windows. The visit to the nearest port is compulsory at the end of the seventh day. The fitness value is obtained by adding a scalar large enough to the symmetric objective function score and using the linear scaling method. Each $|E_i|$-permutation (permutation of $|E_i|$ elements) corresponds to a solution for the subproblem i. In fact, starting at the port of Lisbon at instant 0, the vessel visits all the required fishing stations once, in one of the given time windows, and at the end returns to the port of Lisbon. If the route length is greater than 7 days then at the end of the 7th day the nearest port is visited. Note that all the constraints (3) - (21) are verified with the possible exception of (14) that states the arriving time at Lisbon before the end of day 14. As mentioned above, we expect that the low fitness score of solutions violating (14) will eliminate them from the population in earlier generations.
For the GenRSched genetic algorithm we had tested several known crossover operators suitable for permutations, namely the cycle crossover (CyCx), the order based crossover (OrCx) and the partial match crossover (PMx). After a pre-

liminary computational study we had chosen the PMx operator. As mutation operator we use the swap mutator. We choose to use steady state substitution replacing new% members at each iteration(we also have tested non-overlapping populations and incremental substitution- replacing 2 members of each generation) and the roulette wheel selection method.

Algorithm 3 GenRSched pseudo-code

Require: n, M, pc, pm, new%, maxgen

```
1: Generate an initial population of M random n-permutations.
2: Evaluate the population using the decoder and linear scaling.
3: repeat
4:    Select parents from the population using the roulette wheel selection
      method.
5:    Mate the parents to produce children: to each pair of parents apply, with
      probability pc, the crossover operator PMx and apply to each children,
      with probability pm, the Swap mutator.
6:    Evaluate fitness of the children using the decoder and linear scaling.
7:    Substitute at most new% of the population by the children.
8: until maxgen is reached or population has converged
9: The best solution found is the one corresponding to the highest fitness.
```

Algorithm 4 Decoder pseudo-code

Require: n-permutation

$window = 1$; $pre = 0$; $w_0 = 0$
for $p = n+1, n+p$ **do**
$w_p = 0$
end for
$station = permutation(1)$
$w_{station} = w_0 + \bar{t}_{0\ station}$
$totaldist = t_{0\ station}$
for $k = 2, n$ **do**
$pre = permutation(k-1)$
$station = permutation(k)$
$w_{station} = w_{pre} + prof_{pre} + \bar{t}_{pre\ station}$
if $w_{station} \geq e_{window}$ **then**
if $window = 7$ **then**
$pmin = n+1$
for $p = n+2, n+p$ **do**
if $t_{pre\ pmin} \geq t_{pre\ p}$ **then**
$pmin = p$
end if
end for
$w_{pmin} = w_{pre} + prof_{pre} + \bar{t}_{pre\ pmin}$
$w_{station} = w_{pmin} + prof_{pmin} + \bar{t}_{pmin\ station}$
$totaldist = totaldist + t_{pre\ pmin} + t_{pmin\ station}$
else
$totaldist = totaldist + t_{pre\ station}$
end if
$window = window + 1$
if $w_{station} \geq e_{window}$ **then**
while $w_{station} \geq l_{window}$ **do**
$window = window + 1$
end while
end if
if $w_{station} \leq e_{window}$ **then**
$w_{station} = e_{window}$
end if
else
$totaldist = totaldist + t_{pre\ station}$
end if
end for
$w_{n+p+1} = w_{station} + prof_{station} + \bar{t}_{station\ n+p+1}$
$totaldist = totaldist + t_{station\ n+p+1}$
$objective\_score = totaldist + w_{n+p+1}$
return $objective\_score$ and w_j for all $j \in V$

The sequential heuristic SeqPlus can be summarized in Procedure 5.

Procedure 5 SeqPlus

1: Run GenClusR to obtain two disjoint sets of fishing stations.
2: For each subset of stations, run GenRSched to obtain a solution for the partial SROPs.
3: Obtain a solution to the SROP joining the two partial solutions.

The solutions obtained by the presented sequential approaches can be further improved using ALNS as explained in section 3.3.

3.1.2 Sequential approach based on branch-and-bound

In [7] the SROP is addressed as a particular case of the TSPSTW. Branch-and-bound techniques combined with variable fixing strategies are used to obtain optimal/near-optimal from an efficient standard MILP solver. The proposed heuristic starts with a pre-processing phase where clustering techniques are used to reduce the cardinality of the arc set A. In the second phase, according to the captain guidelines the problem is decomposed into two partial SROPs, one for the north coast and the other for the southwest and south coast. For each partial SROP, a generic MILP solver is then used to obtain a feasible solution. The corresponding linear programming relaxation is solved and whenever the resulting optimal solution is not integer, feasible solutions are obtained by branch-and-bound techniques and variable fixing strategies, within a predefined time limit. In the third phase, the current solutions are analyzed to identify 'good' features that reflect some preferences of the crews as well as operational preferences. These good characteristics are kept by variable fixing before re-running the standard solver, within a predefined time limit. Afterwards, the solutions obtained are improved by the ALNS as explained in section 3.3.

3.2 Integrated Approach - GenShipI

In this section we propose an integrated approach to address the SROP. The goal is to adapt the genetic algorithm GenRSched in order to obtain two disjoint circuits sharing once the port of Lisbon and ensuring that each fishing station is covered once by one of the circuits, while satisfying time windows constraints and the periodic visits to ports. Consequently, the integrated procedure, GenShipI, is similar to phase 2 of SeqPlus but, instead of solving two partial SROP, it considers the set of all fishing stations. In this context, the solutions are represented by permutations of $(n+1)$ elements where n is the number of fishing stations and the additional element, 0, corresponds to the port of Lisbon and is used to identify the end/beginning of each circuit. For example, given an instance of 10 fishing stations, that is $n = 10$, the $(n+1)$-permutation

$$(6, 4, 8, 1, 9, 3, 0, 10, 7, 2, 5)$$

leads to the following partitioning of stations into two circuits, starting and ending at the port of Lisbon:

$$Lis \rightarrow 6 \rightarrow 4 \rightarrow 8 \rightarrow 1 \rightarrow 9 \rightarrow 3 \rightarrow Lis$$

and

$$Lis \rightarrow 10 \rightarrow 7 \rightarrow 2 \rightarrow 5 \rightarrow Lis.$$

The permutation establishes the order in which the stations are visited,starting with the first fishing station to be visited after leaving Lis at time instant $w_0 = 0$. Thus, the genetic algorithm must, mainly, learn how to adjust the two circuits. Given the $(n+1)$-permutation, a decoder similar to the one described in Algorithm 4 determines the starting time of the visit to each station, assuring the validity of time windows. In each circuit, the visit to the nearest port is compulsory at the end of the 7th day. When the decoder reaches the element 0 of the permutation, then the first circuit ends and the second one starts at the 15th day. The procedures continues in a similar way until the end of the permutation is reached. As for the sequential approaches we have no guarantee that the duration of each circuit is equal to or less than 14 days. In fact, the algorithm tend to generate many solutions with one circuit very short and the other with more than 14 days. Therefore, when the constraint (14) is violated, a penalization to the fitness score occurs and the low fitness score of such solutions is enough to rule them out from the population in earlier iterations of the GenShipI algorithm.

The feasible solutions obtained by GenShipI can be further improved using ALNS as explained in the next section.

3.3 Adaptive large neighborhood search

Usually ship routing and ship scheduling applications are associated with a high degree of uncertainty due to bad weather conditions which leads to the demand for robust solutions (see for instance [1]). We propose an ALNS algorithm to introduce some robustness in the solutions obtained by Seq, SeqPlus and GenShipI. Robust solutions offer more stability and flexibility to adapt and recover from disruptions caused by bad weather conditions or setbacks occurring during fishing operations.

Given a solution of the SROP, the objective of the ALNS procedure is to improve the traveled distance in each day, without increasing the number of days at sea.

The proposed ALNS is based on work developed in [10] and [9] by Stefan Ropke and David Pisinger. The ALNS applies several competing removal and insertion heuristics. The selection of a heuristic is based on statistics gathered during the search. At each iteration, the algorithm looks for a better solution by destroying and repairing a part of the current solution. A removal heuristic assigns undefined value to at most q variables followed by a repair heuristic that re-assigns feasible values to those variables.

As noted by several authors the performance of an ALNS algorithm depends on the choice of removal and insertion procedures. One may consider simple insertions heuristics that run in short time. Usually simple and fast heuristics lead to poor quality solutions. However, this disadvantage is offset by the fact that, large moves around the solution space lead to a diverse set of feasible solutions that potentially includes good quality solutions.

Looking to the particular structure of the problem, we noticed two types of routes: daytime routes and nighttime routes. Daytime routes include the set of fishing stations to be visited on a day according to time windows constraints. Nighttime routes connect daytime routes and correspond to the vessel trip from the location of the last visited fishing station on day i and the location of the first fishing station to be visited on day i+1. There are no explicit constraints on nighttime routes however implicitly the survey must be completed in 28 or less days. To exploit this particular structure we have considered four different removal operators.

- Daytime removal - removes a subset of q fishing stations from a daytime route. The subset to be removed do not include the first and the last fishing stations visited on that day in order that nighttime trips, linking consecutive days, remain unchanged.

- Time oriented removal - removes a subset of q fishing stations from daytime route i and from daytime route i+1. The subset to be removed does not include the first and the last fishing stations visited, respectively, on day i and day i+1, so as to not disrupt the connections to days i-1 and i+2.

- Route proximity removal - removes a subset of q fishing stations, which are geographically close, from daytime route i and from daytime route i+k, k> 1. The subset to be removed do not include the first and the last fishing stations of routes i and i+k.

- Nighttime removal - removes two arcs corresponding to two nighttime routes, each arc connecting two fishing stations.

The Daytime removal, Time oriented removal and Route proximity removal operators remove a subset of fishing stations by disconnecting them from their current routes, leaving partial routes and/or isolated fishing stations. By removing two nighttime arcs, the Nighttime removal operator leads to three three partial routes in the solution. The Daytime removal and the Route proximity removal operators only affect daytime routes. The Time oriented removal works in both daytime and nighttime routes. The Nighttime removal moves within the set of nighttime connections. The Time oriented and the Route proximity removal operators are based on the idea that the fishing stations removed from daytime route i are easy interchanged with those removed from day i+k due to geographical proximity, while maintaining time windows constraints feasibility.

At each iteration of the ALNS algorithm, a neighborhood defining operator is selected, among the set of four operators defined above. Since the execution of a removal operator leads to a set of partial sub-routes then, a insertion operator should be applied in order to repair the solution. We have considered that neighborhoods $N_i,\ i = 1, ..., 4$ induced by the removal operators will take care of both the removal and insertion steps. That is, the set of partial sub-routes will be reconnected among them by a least cost insertion operator, ensuring the non violation of time windows constraints. The least cost insertion heuristic looks to routes, that do not include ports, without taking into account it orientation. As a consequence it may reverse the orientation of such type of routes. This can have some impact while repairing nighttime arc removals as illustrated in the example below. In this example, the Nighttime removal operator removes the arcs (6,4) and (9,24). To repair the solution, the least cost insertion heuristic inserts arcs (6,9) and (4,24) by reversing the orientation and swapping days $i+1$ and $i+2$.

Day i: $\rightarrow 23 \rightarrow 1 \rightarrow 8 \rightarrow 6 \rightarrow$	Day i: $\rightarrow 23 \rightarrow 1 \rightarrow 8 \rightarrow 6 \rightarrow$
Day $i+1$: $\rightarrow \mathbf{4} \rightarrow \mathbf{17} \rightarrow \mathbf{19} \rightarrow \mathbf{18} \rightarrow$	Day $i+1$: $\rightarrow \mathbf{9} \rightarrow \mathbf{13} \rightarrow \mathbf{14} \rightarrow \mathbf{16} \rightarrow$
Day $i+2$: $\rightarrow \mathbf{16} \rightarrow \mathbf{14} \rightarrow \mathbf{13} \rightarrow \mathbf{9} \rightarrow$	Day $i+2$: $\rightarrow \mathbf{18} \rightarrow \mathbf{19} \rightarrow \mathbf{17} \rightarrow \mathbf{4} \rightarrow$
Day $i+3$: $\rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow$	Day $i+3$: $\rightarrow 24 \rightarrow 25 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow$

The selection of the operators is based on a roulette wheel scheme. Every removal/insertion operator is associated with a score that represents it past performance at improving the current solutions. Operators that have successfully found improving solutions have a higher score and consequently a higher probability of being chosen. The initialization of the score vector, π, attributes an equal score to each removal/insertion operator. Whenever operator i finds a solution x' better than the current solution x a constant ϵ is added to the correspondent score π_i. The probability of operator i being chosen is given by $\frac{\pi_i}{\sum_{j=1}^{4} \pi_j}$.

The new solution x' is accepted if the corresponding total traveled distance is less or equal then the total traveled distance covered by the current solution x. The ALNS algorithm is summarized in Algorithm 6.

Algorithm 6 ALNS algorithm pseudo-code

Require: x feasible solution

repeat
  Use roulette wheel selection based on the scores vector π to choose a removal/insert operator inducing neighborhood N_i.
  Consider the heuristics correspondent to the induced neighborhood N_i and obtain a new solution x' from x.
  if $vopt(x') < vopt(x)$ **then**
    $x = x'$
  end if
  Update the roulette wheel statistic score π_i of neighborhood N_i
until No improvement over x is achieved within k consecutive iterations

4 Computational experience

We used the GALIB genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology [14] to code the genetic algorithms, GenClusR and GenRSched. We used the R programming language for coding the ALNS procedure. The computational experience was performed on a Intel Core 2 Quad Q6600, 2.4 GHz with 4GB RAM.

4.1 Computational study details

Regarding the genetic algorithms, in each generation, we considered a population size of 50, a probability of crossover of 0.8, a probability of mutation of 0.1 and steady state substitution - replacing 75% members at each generation. The maximum number of generations was set to 5000.

For the computational experience we considered a set of 95 fishing stations and 4 ports namely Lis, Portim, Fig and Matos. Concerning the geographical location of Portugal, 52 fishing stations and 3 ports, Lis, Fig and Matos, are located on the north coast, 23 fishing stations on the southwest and 20 fishing stations and 1 port, Portim, located on the south coast (see Figure 1). Therefore $E = \{1, ..., 95\}$ where fishing stations $1, ..., 52$ are located on the north coast, $53, ..., 75$ are located on the southwest and the remaining $76, ..., 95$ are located on the south. For each fishing station and each port the latitude, longitude and depth are known. We have a forecast of the time spent in the fishing operation at each station, which depends on the respective depth.

We have considered 6 scenarios depending on different weather conditions,

- Very nice weather conditions - the vessel speed is about 11 knots;
- Good weather conditions - the vessel speed is about 9 knots;
- Not so nice weather conditions - the vessel speed is about 7 knots.

and different values for parameter ϵ, which accounts for extra time during the fishing operation,

- $\epsilon = 30$, the estimated value for the current equipment used in fishing operations;
- $\epsilon = 15$, the estimated value for new equipment that might be purchased.

Each procedure, Seq, SeqPlus and GenShipI is run 10 times for each scenario.

4.2 Computational results

In this section we compare from a computational point of view the three heuristic approaches that combine genetic algorithms and ALNS, proposed in sections

3.1.1 and 3.2, as well as the heuristic approach based on branch-and-bound techniques and variable fixing strategies, developed in [7] and presented in section 3.1.2.

Table 2 and Figure 2 show computational results for 10 runs of the GenClusR algorithm executed in phase 1 of both Seq and SeqPlus algorithms. Concerning table 2, row "Min" ("Max") describes the solution with shortest (longest) traveled distance. Row "Average" displays average results for the 10 runs.

	GenClusR		**Solution partition**					
10 Runs	Dist (km)	CPU (sec)	circuit1 stat	# stat visited	Dist (km)	circuit2 stat	# stat visited	Dist (Km)
Min	1782.1	24.9	53 - 95	43	833.3	1 - 52	52	948.8
Average	1821.0	33.0	53 - 95	43	848.6	1 - 52	52	972.4
Max	1915.9	48.8	53 - 95	43	887.4	1 - 52	52	1028.5

Table 2: Computational results for 10 runs of the GenClusR algorithm

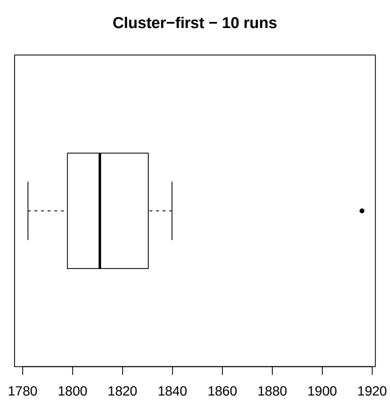

Figure 2: GenClusR algorithm: objective function value for 10 runs

The second and third columns display, respectively, the traveled distance, in km, for the two circuits and the CPU time spent by the algorithm, in seconds, for the 10 runs. Detailed results about each circuit are presented in columns 4 to 9. Columns 4 to 9 show, for each circuit, which fishing stations are visited, the total number of fishing stations visited and the traveled distance. Due to their geographical location, some fishing stations are expected to belong to different circuits. However, there is a group near Lisbon that rose doubts which were clarified through the execution of GenClusR. Concerning the traveled distance one can see from figure 2 that some uniformity in GenClusR behavior as been attained. Each boxplot has only one outlier corresponding to the worst solution obtained over 10 runs of GenClusR.

Tables 3 and 4 report computational results for procedures Seq and SeqPlus, respectively. Twelve instances were considered, each corresponding to a cluster of fishing stations and a scenario.
Table 3 presents results for algorithm Seq followed by the ALNS. Actually, Seq applies the decoder to GenClusR best solution (presented in the first row of table 2). Consequently, the same permutation defines the order by which fishing stations are visited and so, at the end of phase 1, all instances of each cluster have the same traveled distance. However, in phase 2 the greedy algorithm is executed to establish the time windows as well as the visits to the ports yielding, according to the scenario under consideration, different completion times and different traveled distances. Afterwards, the ALNS procedure is applied, to each instance, to adjust the sequence of fishing stations to be visited while improving the traveled distance. It was decided to keep the completion time so as to obtain robust solutions able to absorb some setbacks that may occur on each day.
Table 4 shows SeqPlus+ALNS and SeqPlus results. Tables 3 and 4 present in columns "instance", "Obj Value", "Dist (km)", "Completion time (h)", "# Days visiting", "Max visit/day" and CPU (sec)", respectively, the name of the instance, the objective function (1) value, the traveled distance in km, the completion time in hours, the number of days at sea, the maximum number of fishing stations visited during a day and the time spent by the algorithm, in seconds. The results corresponding to SeqPlus are displayed in the eighth and ninth columns of table 4, respectively, the traveled distance solution and the CPU. It is worth noting that Seq and SeqPlus always produced feasible solutions to SROP.

	Seq + ALNS					
instance	Obj Value	Dist (km)	Completion time (h)	# Days visiting	Max visit/day	CPU (sec)
$N715$	316.2	969	247.0	10.3	6	17.2
$N730$	327.3	1021	254.4	10.6	6	17.6
$N915$	279.3	957	223.0	9.3	7	17.4
$N930$	307.4	982	249.7	10.4	6	17.6
$N1115$	257.1	1011	206.5	8.6	6	17.5
$N1130$	279.0	949	231.5	9.6	6	17.4
$SSW715$	283.8	1071	207.3	8.6	6	13.5
$SSW730$	293.1	858	231.8	9.7	6	13.8
$SSW915$	265.2	1070	202.3	8.4	6	13.4
$SSW930$	267.7	888	213.5	8.9	6	13.7
$SSW1115$	235.9	1023	184.8	7.7	7	13.6
$SSW1130$	255.7	1070	202.2	8.4	6	13.5

Table 3: Best results for Seq followed by ALNS

instance	SeqPlus + ALNS Obj Value	Dist (km)	Completion time(h)	# Days visiting	Max visit/day	CPU (sec)	SeqPlus Dist (km)	CPU (sec)
$N715$	310.3	1123	230.1	9.6	7	17.2	1169	17
$N730$	314.8	1143	234.6	9.8	7	17.2	1169	17
$N915$	265.3	1077	201.9	8.4	8	19.2	1347	19
$N930$	275.5	1091	211.3	8.8	7	17.1	1241	17
$N1115$	244.5	1096	189.7	7.9	8	19.2	1157	19
$N1130$	252.9	1025	201.6	8.4	7	17.2	1025	17
$SSW715$	242.8	853	181.9	7.6	7	15.2	901	15
$SSW730$	251.1	920	185.4	7.7	7	13.2	933	13
$SSW915$	213.7	854	163.5	6.8	8	13.2	854	13
$SSW930$	211.9	828	163.2	6.8	7	13.3	851	13
$SSW1115$	194.9	828	153.5	6.7	7	13.1	828	13
$SSW1130$	202.3	869	158.8	6.6	8	14.1	869	14

Table 4: Best results for SeqPlus and SeqPlus followed by ALNS

Tables 3 and 4 show that it is better to optimize routing and scheduling after the cluster definition (SeqPlus+ALNS) when compared to optimize scheduling after the cluster and route definition (Seq+ALNS). This means that it is worth handling time windows inside the genetic algorithm when designing the route. Concerning the objective function value, which accounts for both the traveled distance and the completion time, SeqPlus+ALNS outperforms Seq+ALNS for all instances. Figure 3 details the behavior of the objective function value throughout the 10 iterations of GenRSched, in phase 2 of SeqPlus. We see three pairs of boxplots, being each pair characterized by the vessel speed. Concerning the distribution of the objective function values one may say that, for the North and SSW instances, boxplots are symmetric in about 4 instances and negatively skewed in 3 instances. The remaining instances have outliers corresponding to the best solution found. This is mainly due to the fact that we have considered only 10 runs. By considering a small number of runs we had in mind to analyze the quality of solutions obtained in short CPU.

Analyzing each component of the objective function value, one can see that Seq+ALNS gives a shorter distance than SeqPlus+ALNS in 7 out 12 instances. This was expected since for Seq the main optimization problem, wich is handled by GenClusR, minimizes traveled distance. These partial results are overshadowed by the fact that, for the completion time SeqPlus+ALNS significantly outperforms Seq+ALNS for all instances. This is also visible in column "Days visiting" which shows that the average number of days at sea for SeqPlus+ALNS is 7.9 while for Seq+ALNS is 9.2 days. Moreover, the largest difference arises for instance SSW930 whose SeqPlus+ALNS solution implies 6.8 days at sea

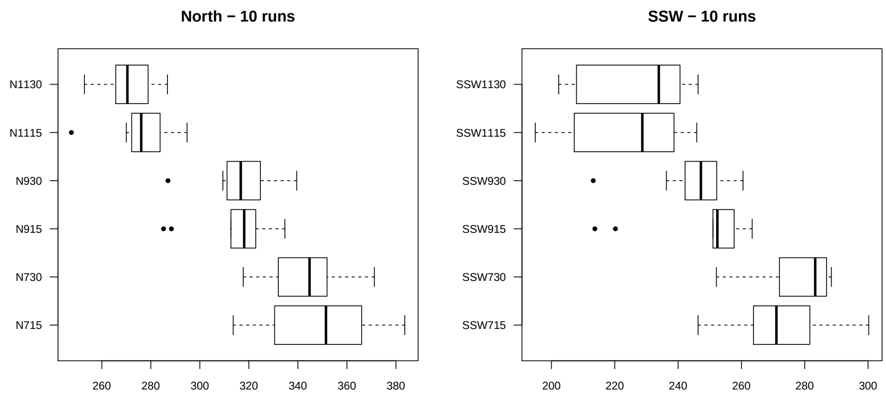

Figure 3: GenRSched: objective function value for 10 runs

while Seq+ALNS implies 8.9 days. The differences between the number of days at sea given by Seq+ALNS and SeqPlus+ALNS may be partially explained by column "Max visit/day". In fact, for Seq+ALNS and SeqPlus+ALNS the maximum number of visits to fishing stations in each day is on average, 6.2 and 7.3, respectively. A larger number of visits per day leads to a shorter number of days at sea and, consequently, a shorter completion time. Concerning the CPU, there are no significant differences between Seq+ALNS and SeqPlus/SeqPlus+ALNS.

As to the question whether it is worth applying or not the ALNS procedure one may noticed that for algorithm Seq the ALNS adjusts the permutation of fishing stations, given by the solution of GenClusR, taking into account the time windows and the compulsory inclusion of the ports established by the decoder, in order to decrease the traveled distance.

Comparing SeqPlus and SeqPlus+ALNS, for the 12 instances tested, one can see that on average the traveled distance decreases by 4.4%, after the execution of ALNS. This decrease takes its maximum value of 20.0% for instance N915 and its minimum value of 0% for instances N1130, SSW915, SSW1115 and SSW1130. Since the increase in CPU for SeqPlus+ALNS is meaningless compared to SeqPlus CPU, one may consider it worth running the ALNS, as a final step, to improve the solution given by the genetic procedure.

Table 5, which follows the same format as Table 3, shows the solutions obtained by the sequential approach based on branch-and-bound and variable fixing. A time limit of 1600 seconds was imposed for the second and third phases of the heuristic. The resulting feasible solutions are further optimized by the application of the ALNS described in 3.3.

From tables 4 and 5, one can see that, concerning the objective function value, the solutions given by the Varfix&Branch-and-Bound+ALNS are all of poorer quality than the solutions given by the SeqPlus+ALNS, or even than SeqPlus. Regarding the traveled distance Varfix&Branch-and-Bound+ALNS had a better performance than SeqPlus+ALNS in 4 out of 12 instances. However,

instance	Varfix&Branch-and-Bound + ALNS					
	Obj Value	Dist(km)	Completion time(h)	# Days visiting	Max visit/day	CPU (sec)
$N715$	336.9	1089	259.2	10.8	6	3205
$N730$	381.5	1141	300.0	12.5	6	3205
$N915$	293.2	1028	232.8	9.7	6	3206
$N930$	312.3	1149	244.8	10.2	6	3205
$N1115$	256.7	957	208.8	8.7	6	3205
$N1130$	293.0	1060	240.0	10	6	3204
$SSW715$	295.9	885	232.8	9.7	6	3205
$SSW730$	320.7	928	254.4	10.6	5	3204
$SSW915$	241.5	883	189.6	7.9	6	3205
$SSW930$	264.3	986	206.4	8.6	6	3203
$SSW1115$	231.2	928	184.9	7.7	7	3204
$SSW1130$	234.4	897	189.6	7.9	6	3205

Table 5: Best results for Varfix&Branch-and-Bound followed by ALNS

the completion time for SeqPlus+ALNS solutions is better than the completion time for Varfix&Branch-and-Bound+ALNS for all the instances. Consequently, the number of days at sea, which depend on the completion time, is always larger for Varfix&Branch-and-Bound+ALNS than for SeqPlus+ALNS. For Varfix&Branch-and-Bound+ALNS, the maximum number of fishing stations visited on a day is on average 6. On average, the CPU time is 6409.3 seconds for Varfix&Branch-and-Bound+ALNS which is clearly higher than the average CPU time of SeqPlus+ALNS, 17.8 seconds.

Table 6 presents the computational results for the integrated approach, GenShipI. In order to compare these results with those obtained for the sequential approaches we associate 3 rows with each instance, the first displays the instance results and below it, two rows corresponding to the detailed results for each circuit. For each row, column '# Fish Stat' presents the number of fishing stations visited and for each circuit, this number is split by region, north + south and southwest. The information displayed in the remaining columns is similar with the one displayed in Tables 3, 4 and 5. Note that, the ALNS procedure adjusts the sequence of fishing stations to be visited in each circuit while minimizing the total traveled distance of the survey. Since there is a significant reduction in the traveled distance, visits to ports might be out of place for the modified circuits after ALNS. In such cases the decoder is applied to the modified circuits redefining the starting time of each fishing operation and the compulsory visits to ports. Despite the large improvement on the traveled distance obtained with the ALNS procedure, on average 18.9%, the results for GenShipI are poor for all instances. In fact, the traveled distance, the completion time, the number of

instance	GenShipI + ALNS						GenShipI	
	# Fish Stat	Dist (km)	Completion time(h)	# Days visiting	Max visit/day	CPU (sec)	Dist (km)	CPU (sec)
$NSSW715$	95	3041	-	23.5	6	859.0	3564	858.7
$NSSW715\_1$	29+17	1746	283.2	12.7	6	-	2005	-
$NSSW715\_2$	23+26	1294	282.9	10.8	6	-	1559	-
$NSSW730$	95	2659	-	20.7	6	808.5	3146	808.1
$NSSW730\_1$	41+0	1199	255.7	8.9	6	-	1344	-
$NSSW730\_2$	11+43	1460	305.4	11.8	6	-	1802	-
$NSSW915$	95	3139	-	21.6	7	811.3	3651	811.0
$NSSW915\_1$	43+5	1917	261.5	11.1	7	-	1822	-
$NSSW915\_2$	9+38	1222	259.4	10.5	6	-	1829	-
$NSSW930$	95	3175	-	19.9	6	793.0	4224	792.5
$NSSW930\_1$	33+9	1586	227.3	9.2	6	-	1820	-
$NSSW930\_2$	19+34	1589	311.8	10.7	6	-	2404	-
$NSSW1115$	95	2572	-	18.3	7	809.9	3484	809.5
$NSSW1115\_1$	42+9	1343	232.1	8.8	7	-	1878	-
$NSSW11152$	10+34	1229	233.5	9.5	7	-	1606	-
$NSSW1130$	95	3396	-	18.8	7	794.1	4107	793.8
$NSSW1130\_1$	34+8	1746	255.5	9.1	5	-	2051	-
$NSSW1130\_2$	18+35	1650	234.2	9.7	7	-	2056	-

Table 6: Best results for GenShipI and GenShipI followed by ALNS

days at sea are much larger than the corresponding results for all the sequential approaches. CPU time is also much larger than the one obtained by the sequential approaches Seq+ALNS and SeqPlus+ALNS. In spite of the possible benefits of integrating all the decisions, the proposed integrated approach is clearly outperformed by all sequential approaches. Handling the whole SROP instead of two partial SROPs results in larger problems to be optimized by the genetic algorithm. Although we believe that the dimension issue is mainly responsible for a large CPU, we think that the explanation of the poor results is intrinsic to the genetic algorithm design. For the sequential approaches, the compulsory visit to the nearest port, at most one per circuit, has been well 'understood' by the genetic algorithm, matching well with the chosen codification and fitness evaluation, guiding the algorithm towards promising solutions. On the other hand, when we consider the whole set of fishing stations, dealing in a similar way with the compulsory visits to the ports, the genetic algorithm had difficulty to balance this issue with the minimization of the completion time as well as with the allocation of the visit to the port of Lisbon at the end of the first circuit. The resulting solutions in table 6, despite the improvement, are poor and have large "geographic jumps". The ALNS procedure was able to eliminate some of these jumps but this was not enough to obtain good quality solutions. Figure 4 shows the solutions given by SeqPlus(+ALNS) and GenShipI(+ALNS) for the instance NSSW915. One can visually assess the difference in the quality of the solutions produced by both methods. The improvement obtained by the ALNS procedure in the GenShipI solution is clear but it is also obvious that the resulting solution is poorer than SeqPlus+ALNS solution or even than SeqPlus solution.

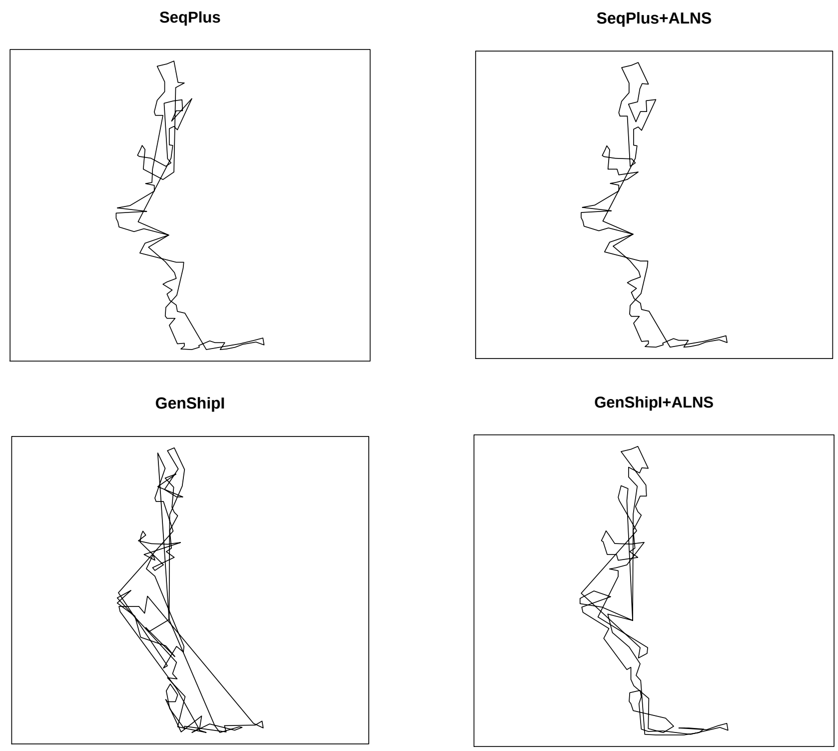

Figure 4: Comparing solutions: instance NSSW915

We believe that the codification of the solutions and the fitness evaluation did not match. If so, to obtain better solutions a different codification and/or fitness evaluation should be considered.

In order to assess the quality and applicability of SeqPlus+ALNS solutions, we have analyzed the route followed by the ship in previous surveys. We noticed that in real situations the maximum number of fishing stations visited in one day ("Max visit/day") was about 6, which is less than the corresponding value given by SeqPlus+ALNS for all scenarios. For such days, we saw that in real solutions the ship traveled distance was greater than the traveled distance given by SeqPlus+ ALNS solutions. This improvement in the traveled distance allows to visit more fishing stations, in one day and, consequently, it also allows that the all survey is completed in less days.

With regard to the two different values that were considered for the parameter ϵ, for $\epsilon = 15$, associated with new equipment, algorithm SeqPlus+ALNS points to an average saving of 5.7 hours in the completion time of each circuit. This average savings goes up to 18.7 hours for the Seq+ALNS algorithm and rises 21.2 hours for Varfix&Branch-and-Bound+ALNS algorithm.

5 Conclusions

In this paper, optimization techniques are used to determine the route of a research vessel that performs a sampling survey to study the abundance of fish species. Environmental concerns and management and staff preferences combined with specific problem requisites concerning the duration of the route and time windows for the sampling survey in each fishing station guided the search for SROP solutions. The goal was to obtain solutions that minimize the traveled distance, to reduce fuel consumption and the consequent environmental pollution related to environmental emissions and sea pollutants, and minimize route completion time, a management and staff preference. We present a mathematical formulation that highlights three combinatorial subproblems included in SROP, namely a clustering, a routing and a scheduling subproblem. To solve SROP and taking into account this underlying structure, we propose two sequential approaches, Seq and SeqPlus, both using a two phase algorithm. In Seq a genetic algorithm, GenClusR, solves the clustering and routing subproblems in phase 1 and a greedy procedure is used in phase 2 to solve the scheduling subproblem. SeqPlus uses the genetic algorithm GenClusR to define the clusters in phase 1 and, for each cluster, uses the genetic algorithm GenRSched to solve an integrated routing and scheduling subproblem in phase 2. Although Seq and SeqPlus produced good quality feasible solutions for SROP in short CPU time, SeqPlus outperforms Seq, mainly due to the fact that SeqPlus solutions have shorter completion times and similar traveled distances. These results suggest that further integration might be worth investigating leading to the development of a third approach, GenShipI, integrating the three subproblems. Unfortunately, GenShipI yielded very poor quality solutions pointing towards a different codification of the solutions and/or a different fitness evaluation. As a final step we propose an ALNS procedure to improve the quality of the solutions obtained by all the above referred approaches. The ALNS showed to be able to decrease the traveled distance in short CPU providing some robustness to the solutions in order to include daily drawbacks. As a final remark we would like to point out that algorithm SeqPlus+ALNS proved to be a suitable tool to solve SROP under different scenarios which may cover different weather conditions as well as new fishing operations conditions.

Acknowledgments

This research was partially supported by Portuguese National Funding from FCT under project PEst-OE/MAT/UI0152.

References

[1] M. Christiansen and K. Fagerholt, Robust ship scheduling with multiple time windows, Naval Research Logistics 49 (2002), 611-625.

[2] M. Fischetti, J.-J. Salazar-González and P.Toth, The generalized traveling salesman and orienteering problems. In G. Gutin and A. P. Punnen, editors, The Traveling Salesman Problem and its Variants, Kluwer (2002), 609-662.

[3] M. Gendreau, G. Laporte and F. Semet, A branch-and-cut algorithm for the undirected selective traveling salesman problem, Networks 32 (1998), 263-273.

[4] B. Golden, L. Levy, and R. Vohra, The orienteering problem, Naval Research Logistics 34 (1987), 307-318.

[5] J. Karbowska-Chilinska and P. Zabielski, Genetic Algorithm Solving the Orienteering Problem with Time Windows. In Jerzy Switek, Adam Grzech, Pawe Switek and Jakub M. Tomczak, editors, Advances in Systems Science, volume 240 of Advances in Intelligent Systems and Computing, Springer International Publishing (2014), 609-619.

[6] N. Labadie, R. Mansini, J. Melechovsky and R. W. Calvo, The team orienteering problem with time windows: An LP-based granular variable neighborhood search, European Journal of Operational Research 220 (2012), 15-27.

[7] M. Mesquita, A. Murta, A. Paias, and L. Wise, Tsp with multiple time-windows and selective cities. In D. Pacino, S. Voss, and R. Moller Jensen, editors, Computational Logistics, Lecture Notes in Computer Science. Springer (2013), 158-172.

[8] J. Pietz and J. O. Royset, Generalized orienteering problem with resource dependent rewards, Naval Research Logistics 60 (2013), 294-312.

[9] D. Pisinger and S. Ropke, A general heuristic for vehicle routing problems, Computers & Operations Research 34 (2007), 2403-2435.

[10] S. Ropke and D. Pisinger, An adaptative large neighborhood search heuristic for the pickup and delivery problem with time windows, Transportation Science, 40 (2006), 455-472.

[11] F. Tricoire, M. Romauch, K. F. Doerner, and R. F. Hartl, Heuristics for the multi-period orienteering problem with multiple time windows, Computers & Operations Research, 37 (2010), 351-367.

[12] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden, Iterated local search for the team orienteering problem with time windows, Computers & Operations Research, 36 (2009), 3281-3290.

[13] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, The orienteering problem: A survey, European Journal of Operational Research 209 (2011), 1-10.

[14] M. Wall. Galib: A c++ library of genetic algorithm components, version 2.7. Technical report, Massachusetts Institute of Technology (2007).