
A heuristic algorithm searching optimum routes for maritime transportation

Konstantinos Chainas

Athens University of Economics and Business,
15° Km Chalkida – Anthidona,
34100, Loukissia Evias, Greece
Fax: +30-2221061982
E-mail: xainas@otenet.gr

Abstract: The present essay presents a heuristic algorithm for the optimisation of coastal vessel's routes. The task is to research the optimum connection between two ports, incorporating in their routes intermediate nodes, which must satisfy certain limitations and some preconditions related to total distance, trip duration, and demand coverage. This issue has many common elements with the overall vehicle routing problem and the team orienteering problem, but, at the same time, presents with significant particularities. Its particularities consist mainly in the fact that in the sea area, each node may be theoretically linked to any other node, without limitations, while a road network has a specific structure. The results of the algorithm implementation for its region of application were improved compared to that in the literature. The advantages of its application by the maritime companies for the search of the optimal itineraries will be important mainly in terms of resource saving.

Keywords: optimisation; maritime; network; heuristic; algorithm; search.

Reference to this paper should be made as follows: Chainas, K. (2016) 'A heuristic algorithm searching optimum routes for maritime transportation', *Int. J. Computational Complexity and Intelligent Algorithms*, Vol. 1, No. 1, pp.35–53.

Biographical notes: Konstantinos Chainas is a PhD candidate in the Department of Informatics at Athens University of Economics and Business. He is a Mathematician with postgraduate studies in the Department of Marketing and Communication at Athens University of Economics and Business. He works as a Programmer Analyst on ERP systems. His research interests focus on integrated business information systems and decision support systems. He has published the book *Basic Issues on ERP Systems* (GKIOURDAS Publications, Athens).

1 Introduction

The optimum vehicle routing problem (VRP) (Christofides et al., 1979), with regard to the coverage of demand of an area, under specific time limitations and resources, is a non-polynomial (Desrochers et al., 1992) and 'hard' (Savelsbergh, 1985), issue and is included in the category of problems of routing with time and capacity limitations ('vehicle routing problem with time windows – VRPTW') (Solomon, 1987). Given that

in our problem, that is coastal vessels' routing for the coverage of the demand of an area, the starting point and the terminal point do not always coincide, we could say that the coastal vessels routing problem is closer to team orienteering problem (TOP) of multiple vessels (Tang and Miller-Hooks, 2005). As per Tang and Miller-Hooks (2005), the methodology for searching optimum solutions in TOPs is different than the one used in VRPs. According to Toth and Vigo (2002), the scope of vehicle routing under time and capacity limitations is to minimise routing costs and to serve customers. In general, in order to handle this problem category, there has been development, in the recent years numerous post-heuristic, approximation algorithms, which provided quite satisfying solutions. Laporte and Osman (1995) and other researches claim that the satisfactory solution of the VRP is provided with the use of heuristic and post-heuristic methods. Particularly satisfying solutions have also been provided in TOP using post-heuristic methods, and, in particular using the tabu search algorithm (Cordeau et al., 2001). Tang and Miller-Hooks claim that, for some reasons not theoretically justified yet, tabu algorithm provides better solutions compared to other approximation algorithms. On the basis of the prior experienced gained over the past decades on routing problems in general (Vidal et al., 2011), and specifically for the coastal vessels routing problem, we developed an approximation algorithm which has many common points of reference with tabu search method, as well as with other approximation, constructive, heuristic and post-heuristic algorithms. The proposed heuristic algorithm with the symbolic name 'NAUTILOS' which shall be presented below, provides improved solutions with regard to existing routes and the structure of the current network, as well as with regard with the overall time-distance covered by all routes. The algorithm results' comparisons were performed using all data found in related bibliography (Aifantopoulou, 2004) and with those found in SETHAM Project-Ministry of Merchant Marine of Greece (2001). The contribution of this essay is that, additionally to the search of optimum routes in the sea area, this algorithm can also be applied in general VRPs providing very good results. Moreover, the algorithm can also be applied in network design problems, where optimum connection between two different network nodes under certain criteria is desired. The good behaviour of the algorithm in all such routing and network design problems is attributed, in our opinion to the very structure and operation of the algorithm, which, essentially 'constructs' the optimum solutions step-by-step, based on the limitations set and avoiding local optima in an 'intelligent' way.

In essence, the problem of optimising the maritime transportations that we deal with in this report is a TOP problem, since we want to design optimal itineraries by covering adequately the demand of a specific region.

2 Problem description

Assume a sea region consisting of n ports (nodes) $p_i, i = 1, 2, \dots, n$. Assume, we avail k vessels with different features (capacity, speed, etc.). We have one departure port for each route and one terminal port, which in general differs from the departure port. Each route has a constant frequency over a specific time period (i.e., 7 times a week), a specific time of departure and specific time window for serving each node. That is, each vessel has a specific time window for the departure and arrival at each port. Each port (node) in the search area belongs to at least one, or more routes. A maritime company is interested in routing the vessels of its fleet on the routes presenting with the maximum profit. Thus,

the objective is to search the optimum routes which, within set time limits, shall assimilate the types of vessels available and shall cover the demand of the port-nodes of the required sea region to be covered, at the lowest cost possible.

3 Basic concepts and assumptions

- 1 Port of departure (p_d), is the port of departure, starting of the route.
- 2 Port of arrival (p_a), is the port at which the route terminates, which, in general, is different from the port of departure.
- 3 Node-port ($p_i, i = 1, 2, \dots, n$) is an intermediate port of a route, other than the departure and arrival ports.
- 4 Wait time of a vessel is the time that a vessel remains in an intermediate port for disembarkation – embarkation (t_w).
- 5 Total time is the time of the route (t_s).
- 6 Search area (U_x) is the area in which the nodes (ports) that shall compose a route are to be searched. This area may be part of the sea region (i.e., Cyclades), for which we wish to design optimum routes, or the entire sea area (i.e., Aegean Sea). This area consists of various regions – neighbourhoods (i.e., Eastern Cyclades), in which we apply the prohibition or release policies, depending on the data. That is, during the search process, it is possible for a node of a distinct region (neighbourhood), though included in the prohibition list due to limitations, to be ultimately improving the value of the objective function, rendering thus its release and integration in the route necessary and vice versa.
- 7 Value of connectivity of three ports (a, b, c) (Figure 1). This parameter is particularly important for the algorithm and it is the element that differentiates it from standard optimisation algorithms. The basic concept for introducing this parameter in the algorithm was created from the need to find a mathematical way to introduce a third port between two other ports that we wanted to link. Assume two ports a and b which have a distance of d_{ab} between them and assume that we want to introduce between them a third port, c, which has a distance of d_{ac} from a and a distance of d_{cb} from b, which is going to be the criterion for the incorporation or dismissal of the third port. This is the connectivity value and is expressed as the quotient of the division $(d_{ac} + d_{cb})/d_{ab}$ which is called connectivity value. This value is compared with the values of the interval [1, 1.5], depending on whether we wish a greater, or smaller, deviation from the course defined by straight line ab. This interval of values was selected after many tests of the algorithm on the actual distances between the ports of the Aegean Sea. Thus, if the connectivity parameter assumes a value of 1, then we have the so called, inelastic connectivity and in this case route $ac \rightarrow cb$ does not deviate at all from route ab, while, if it assumes value 1.5 we have the so called elastic connectivity and in this case, route $ac - cb$ deviates more from route ab. Obviously, in the 1st case there is only inclusion of the ports located on straight line ab, while in the 2nd case, many more ports are included. We assume that the

algorithm is applied for small sea distances (coastal links), because for much longer distances, it would be necessary to use sea arcs.

Figure 1 View of two ports a, b and intervention of a third port, c, in accordance with the connectivity value

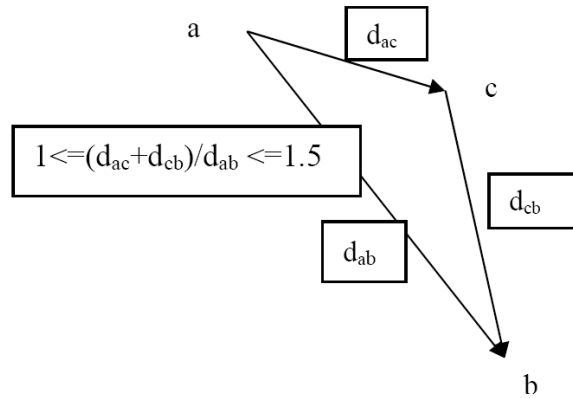
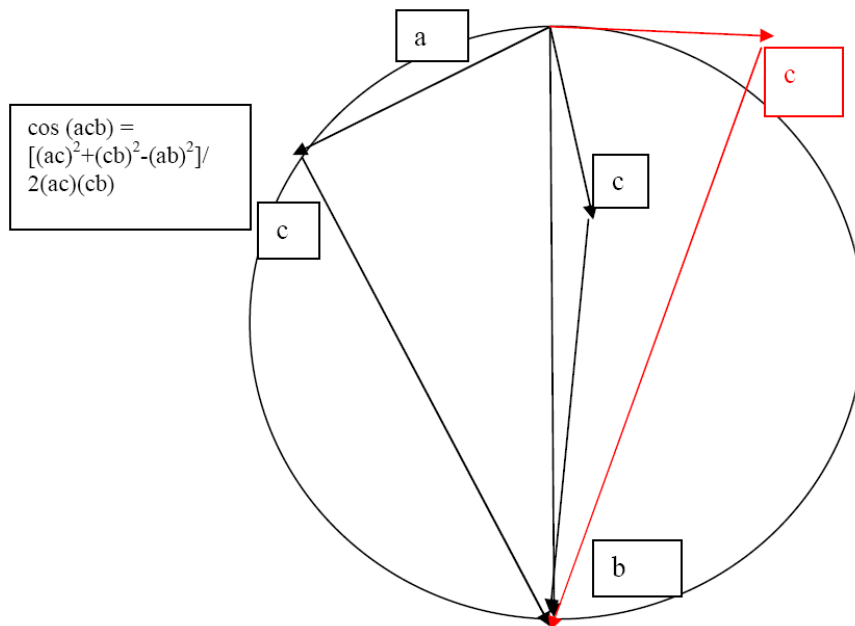


Figure 2 The cyclical search area of intervening ports having a search angle of 90° (see online version for colours)



- 8 Sweep angle is the angle created by the points of the two ports a and b following the interposition of port c, that is, angle acb. Sweep or search area is the area of the cylindrical disc created each time during the process of connecting two ports a, b and interposing the third port c. The cylindrical disc in question has as its diameter distance d_{ab} , that is the distance between the two ports a and b. The third port shall be searched in the cyclical area created, having as its centre the middle of the distance d_{ab} and as its radius $d_{ab}/2$. This limitation provides a direction to the route from port a to port b, given that it dismisses all connections outside the said cyclical area and prevents 'back turns' of the routes.

Figure 3 The elliptical (A) search area of intervening ports with a search angle smaller than 90° (see online version for colours)

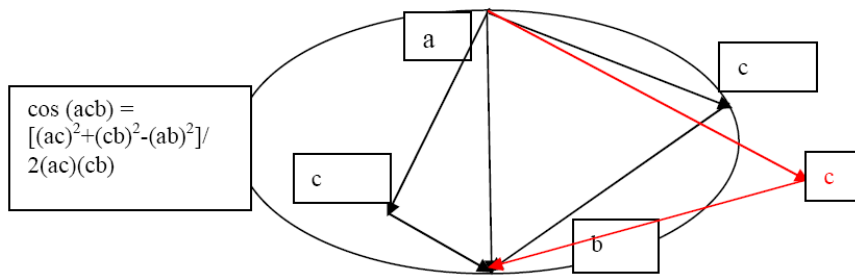
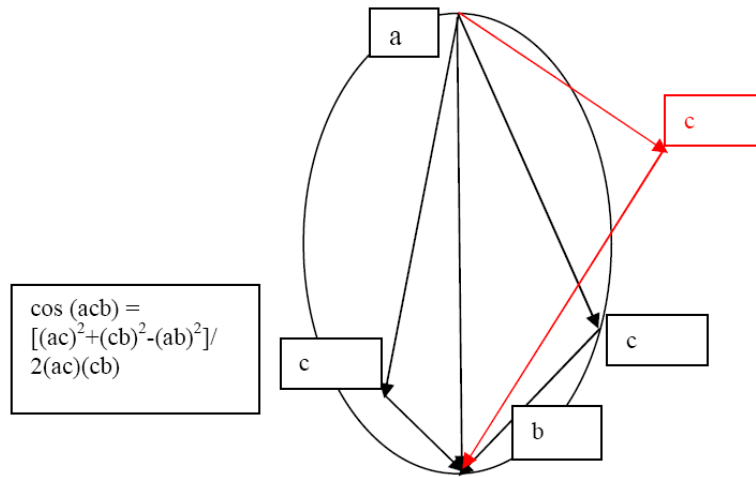


Figure 4 The elliptical search area of intervening ports with a search angle greater than 90° (see online version for colours)



Applies: $\cos(\text{acb}) = \frac{[(ac)^2 + (cb)^2 - (ab)^2]}{2(ac)(cb)}$, where $\cos(\text{acb}) \leq 0$.

This value ensures that port c is within the cyclical area created on the basis of distance ab (sweep or search area). The values that can be assigned to the sweep angle are the values within interval $[90^\circ, 180^\circ]$ that ensure that port c is located in the region of the cyclical disc having diameter ab (Figure 2). In some cases, angles smaller than 90° can also be accepted, when it is desired to broaden or narrow down

the search area, which is formulated on the basis of the departure and final destination ports. In this case, there is formation of elliptical search areas (Figures 3 and 4).

- 9 A release strategy is followed when we release a node or subset of nodes from the prohibited list, in order to integrate them in a route, because they improve a previous solution.
- 10 A prohibition strategy is followed when we prohibit one or more nodes from participating in a route, at some point during the search. It should be noted that such prohibition may be raised under certain conditions.
- 11 Prohibited list. This list is populated dynamically during the search. The initial list includes all nodes, except for the departure and termination nodes, for which the values of the sweep angle and connectivity are not within the predefined limits. During the search process of the algorithm and depending on the limitations and the value of the objective function, we follow release, or prohibition strategies. That is we release certain nodes from search list and include in it some other nodes, depending on the case. It is possible for a node on a route having the same departure and destination, to be integrated in a route and not to be integrated to another, when the optimum routes searched on the basis of the search conditions are different.
- 12 Ambition criterion consists in the following. As long as a node during the search process provides a better value for the objective function, then, prohibition can be raised and the node can be integrated in the route. For instance, if connectivity value or the sweep angle prohibits the participation of the node in the search process, if its participation provides a better value for the objective function, compared to the past, then we can raise prohibition and allow participation of the node on the route. It is possible, in case the participation of a node included in the prohibited list leads to an improved route compared to the routes created up to that point, to release the node and integrate it to the route.
- 13 Search time. Depending on the case, it is possible to set some rules. These may be static, that is invariable, and in other cases we may need dynamic review, that is to be able to modify the duration of a route, i.e., in case of emergency conditions. In the model applied for the islands of the Aegean, we have set as the upper limit of the duration of a route, a value of 24 hours. This limitation is placed because the short distance itineraries do not last longer and this is also a terminating term of the algorithm.
- 14 Search algorithm memory. In this memory there is storage of all information necessary, or created during the search process, in order to improve the routes generated. This memory shall store the information concerning prohibitions, releases, routes generated and any other information produced during the search progress. Actually, this process may require some extra time and extra memory resources, but it ensures non-recycling of the solutions.

- 15 Procedure of sorting during search. The sorting required to be performed during the search process depends on the factors defined as predominant ones. For instance, if distance is set as a predominant improvement factor, then this shall be necessary each time we look for the next move in a sub-region – neighbourhood, and shall be searched in the nearest neighbours – nodes of the last node integrated in the route. In this context, sorting by distance is necessary.
- 16 Search stop conditions of the algorithm, are the conditions which, if one of them applies, the algorithm stops. These are:
 - a the total duration of the route, which in the specific application of the algorithm cannot exceed 24 hours
 - b the maximum vessel capacity, which may not be greater than the total demand of the route nodes
 - c the maximum search time of the algorithm, that is, if an upper limit for search duration is set.

In this particular application for coastal routes, we have set the first two stop conditions in effect.

4 Cost parameters for the participation of a node in a route

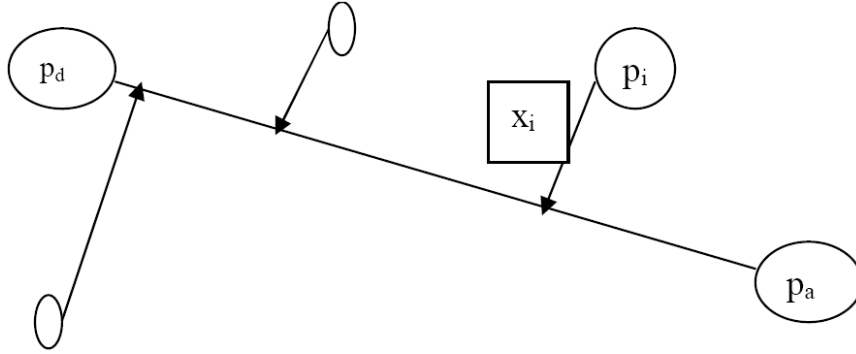
Routing problems are generally evaluated using two methods. The first one refers to costs minimisation (Louis et al., 1999). The second refers to the profit (profit or reward) maximisation (Tang et al., 2010). In essence, these constitute the same methodology, given that inverting cost coefficients the problem is converted from cost minimisation to profit maximisation.

On the basis of this logic, below follows an analysis of the cost parameters comprising the objective function of the algorithm:

- 1 The total distance of the route $\Sigma d_i, i = 1, 2, \dots, n$. Each node j added to the route after the last node i ‘burdens’ the total distance travelled of the route per $d(i, j)$. That is, $\Sigma d_i = D_j + d(j, i)$ where $i, j = 1, 2, \dots, n$ (this parameter can also be referred to as duration, if we convert distance to time duration, using formula $v = s/t$, and assuming $v =$ average vessel speed, $s =$ interval travelled and $t =$ time). It consists of the various distances among nodes.
- 2 The total route time $\Sigma t_i, i = 1, 2, \dots, n$. This time consists of the sum of the times for the transition from each node of the route to the next (t_{ij}). That is, there is no inclusion of the times spent at each node for the embarkation and disembarkation process (service time).
- 3 The total waiting time of each vessel at each node for the embarkation and disembarkation process, $\Sigma w_i, i = 1, 2, \dots, n$. This time at each node has a specific start time and a specific end time. This time interval is also called time window. If a vessel reaches node i prior to time period w_i this is not attributed to something and it is responsible to wait till the time period for the commencement of the time window, and if it reaches the node later than the start time, then time counts as of that moment, or we can set a type of penalty.

- 4 Total vessel overload $\Sigma o_i, i = 1, 2, \dots, n$. This value is the positive difference between the demand at each node m_i and the capacity of each vessel $c_k, k = 1, 2, \dots, K$. If this value is close to zero, then demand is covered in an ideal manner. If the absolute value of this difference increases, then vessel capacity is lower than the demand and the vessel is not able to cover it, or vessel capacity is much greater than the demand, and in each case there is an error in the selection of the routing vessel type.
- 5 The sum of the distance of each node from the straight line that connects each node with its projection to the straight line that connects the departure node with the terminal node, $\Sigma x_i, i = 1, 2, 3, \dots, n$ (Figure 5). This parameter further burdens the value of the function with the participation to the route of nodes that are quite distant from the straight line connecting the departure and termination nodes.

Figure 5 Vertical distances of the nodes from the axis that connects the departure node with the terminal node



That is, the specific cost parameter ‘favours’ nodes located at close distances from axis $p_d p_a$ and ‘discourages’ the participation of nodes located far from this axis.

The cost of each route r_i is the value of the function:

$$F(r_i) = \lambda_1 \Sigma d_i + \lambda_2 \Sigma t_i + \lambda_3 \Sigma w_i + \lambda_4 \Sigma \sigma_i + \lambda_5 \Sigma x_i, i = 1, 2, 3, \dots, n \quad (1)$$

where

Σd_i the total distance of the route

Σt_i the total time of the route (excluding the wait, embarkation, disembarkation times of the vessel at each node)

Σw_i the total delay time at each port (that is the wait, embarkation, disembarkation times of the vessel at each node)

Σo_i the total overload (the positive difference of vessel capacity and demand)

Σx_i the total distance of each node from the straight line that connects the departure node and the terminal node.

And $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5$ are the gravity coefficients for each cost factors, assuming values that are determined according to the criticality and weight attributed to each cost factor, depending the routing problem and the prioritisation provided each time. We can

introduce the concept of prevalence of a cost factor and define the set of optimum solutions which shall be subject to the dominance relation established. In the problem examined in the essay, that is, the search of the optimum connection between two nodes, we have a series of optimisation criteria, such as total distance, total time and demand coverage. The comparisons with the routing cases referenced in the bibliography, refer to the total distance and on the basis of this criterion we created, at an initial stage, a set of optimum routes for the required connections. Then, at this set of routes, we can apply the other optimisation criteria defined, in order to create final optimum solutions on the basis of an optimisation based on multiple criteria optimisation process. And this is because a maritime company is not only interested in designing the optimum routes for a destination, but its primary concern is whether such routes can be covered using the available fleet and available types of vessels. This is the exact object of the second phase of routes optimisation. There is search of the adequate types of vessels for the coverage of the demand of the destinations of each route created during the previous optimisation phase and vice versa. On the basis of the available vessels of the company, we search which of the optimum routes of the optimisation process can be covered.

5 Mathematical model of the problem

As stated above, the problem of routing vessels with time and capacity limitations (such as the coastal routes problem), is a difficult problem with non-polynomial solution, in the context that using the standard optimum route search algorithms, its solution becomes very difficult and involves exponential times. Using these data, we converted the problem so as to acquire a polynomial solution.

Assume a graph, $G(N, A)$ where N are all the nodes (ports) and A is the total of distances d_{ij} (edges or arcs), of the connections of all nodes (i, j) between them. A node j , in order to be integrated in a route, has a cost related to certain cost parameters (distance, etc.), as analysed in cost parameters. The route that corresponds to each vessel is a path of given direction from the departure node to the terminal node (terminal port), that serves a specific set of intermediate nodes (ports). Each vessel has a given loading capacity (tonnage) and there is a pre-determined demand for each node (port). The sum of the demand of all nodes cannot exceed the total capacity of the vessel which is set for the specific route. The service provided to each port (node) i , must be materialised within a specific time interval (time window) $[w_{1i}, w_{2i}]$, where w_{1i} is the earliest time point and w_{2i} the latest one, within which access to the node (port) must be performed. If the vessel reaches the node (port) earlier, it must wait till it is within its duty time $[w_{1i}, w_{2i}]$.

Moreover, the total routing time (trip time) must be within a framework, i.e., [0.24 h]. Thus, the departure time for each destination must be specific and in conjunction with the time of arrival at each node and at the terminal node, so that it is never in conflict with the operation of the ports, as well as any other limitations that may have been set, such as the ability to access a port at specific times, etc. Thus, the problem is expressed as follows. As long as we avail a specific number of vessels (fleet of vessels), the objective goal is to minimise the total cost of a set of routes that satisfy the requirements of the problem of serving the nodes (ports) of set N . That is, we shall design a set of routes of minimum cost, one route per vessel, so as to cover the demand of each node (port). If we want to limit the number of vessels we can set a cost at the entry of a new vessel. Thus,

we can minimise the number of the fleet on the basis of the cost of introducing a new vessel. The vessels routing problem in its most simple version, and setting the appropriate values at the cost parameters of equation (1), can be converted to a minimum path problem. That is the search of the shortest path between two nodes. However, in this case too, if nodes are n , then the various paths are $1/2 n!$. For instance, for $n = 20$, that is as many as the greatest ports of Cyclades, approximately, then there are $1.2 e^{18}$ different routes!.

In order to formulate the objective function, we define the following:

- P_d , the departure node (departure port)
- P_a , the terminal node (terminal port)
- $p_i, i = 1, 2, \dots, n$ the intermediate ports (nodes) of the search area
- d_{ij} , the distance of node j from node i
- k_i , the total number of vessels $i = 1, 2, \dots, k$
- q_i , the tonnage of each vessel, $i = 1, 2, \dots, k$
- N , the ports in the search area $\{1, 2, 3, \dots, n\}$ without inclusion of the departure and termination ports
- m_i , the demand at each port
- c_{ij} , the costs of connecting two ports i and j
- t_{ij} , the time of each connection of port i with port j
- Each route starts from the central departure port
- The number of the final optimum routes equals the number of the vessels.
- The tonnage of the vessels covers the demand of the ports (nodes) served.
- Each vessel on a route, visits each port (node) of the route only once.
- The wait time of the vessel at each port (node) $w_i, i = 1, 2, \dots, n$ is the vessel's service time and the objective is to minimise it. This time can differ for each node. In the algorithm, for the routes created and for the comparisons, we set a $w_i, i = 1, 2, \dots, n$ value equal to $1/4$ of an hour, which constitutes a reasonable wait time for a vessel at a port for the disembarkation and embarkation of vessels and passengers.
- Each route must be completed within a specific time (≤ 24 hours)
- We set the decision variable $X = \{x_{ijk}\}$ and $i, j = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$ with $i \neq j$.
With $x_{ijk} = 1$ if vessel k connects port i with port j .
And $x_{ijk} = 0$ otherwise.
- We also set $Y = \{y_{ik}\}$ with $i = 1, 2, 3, \dots, N$ and $k = 1, 2, \dots, K$.
With $y_{ik} = 1$ if vessel k visits node i .

And $y_{ik} = 0$ otherwise.

- The problem is to minimise the objective function

$$F = \sum_{i=1}^N \sum_{k=1}^K f_i y_{ik} \quad (2)$$

- Limitations

- a K vessels depart from the central port of departure

$$\sum_{i=1}^N \sum_{k=1}^K y_{ik} = K \quad (3)$$

- b One vessel arrives to node j from node i

$$\sum_{i=1}^N x_{ijk} = y_{jk} \quad \forall k \in K, i, j = 1, 2, \dots, N \quad (4)$$

- c One vessel departs from node i and is directed to j

$$\sum_{j=1}^N x_{ijk} = y_{ik} \quad \forall k \in K, i, j = 1, 2, \dots, N \quad (5)$$

- d Each node participates only once in a route

$$\sum_{k=1}^K y_{ik} \leq 1 \quad \forall k \in K, i = 1, 2, \dots, N \quad (6)$$

- e From the departure node, each vessel departs once

$$\sum_{k=1}^K y_{ik} = 1 \quad \forall k \in K, i = 1, 2, \dots, N \quad (7)$$

- f The vessels on each route are not allowed to move on circular paths

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq \sum_{i \in S} y_{ik} - 1 \quad \forall S \subseteq N, \forall k \in K, S \neq \emptyset, i = 1, 2, \dots, N \quad (8)$$

- g The tonnage of each vessel is lower than or equal to total demand

$$\sum_{i=1}^N y_{ik} m_i \leq c_k \quad \forall k \in K, i = 1, 2, \dots, N \quad (9)$$

- h The total time of each route must be less than the defined $T_{\max} = 24$ h

$$\sum_{i=1}^N y_{ik} t_i \leq T_{\max} \quad \forall k \in K, i = 1, 2, \dots, N \quad (10)$$

- i As long as the connection of node i to node j is integrated in the route, then the starting time for the service of node j must be greater than or equal to the starting time for the service of node i plus the time of service of node i plus the time for the transition from node i to node j .

$$x_{ij}(t_i + w_i + t_{ij}) \leq 0 \quad \forall i, j = 1, 2, \dots, N \quad (11)$$

- j The time of service of node j must be within the defined time window. That is, if the vessel reaches node j earlier than the start of its time window, then the vessel must wait till its time window opens, while if it arrives later, service shall begin from the time of arrival.

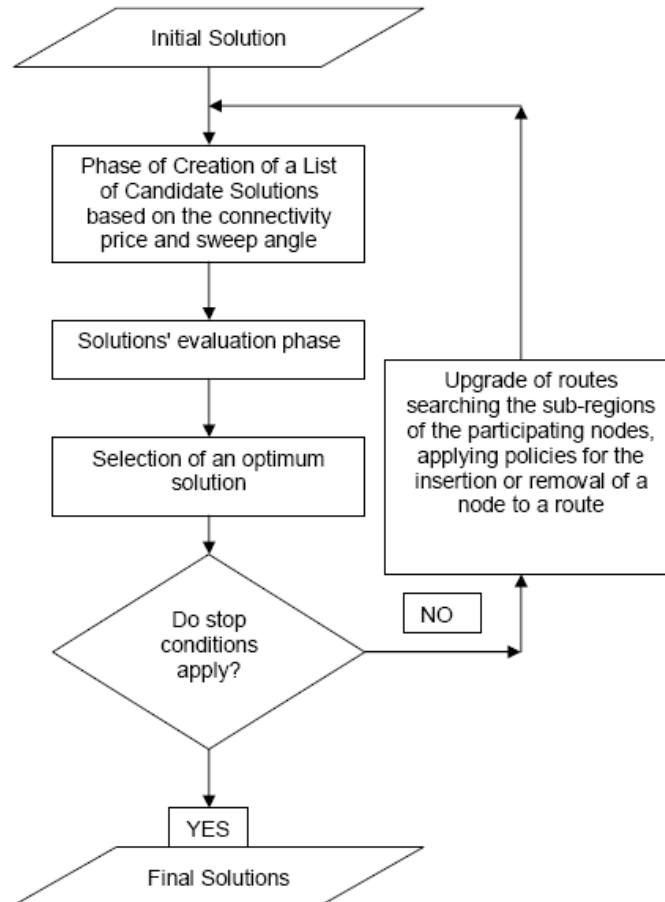
$$w_{1j} \sum_{i=1}^N \sum_{j=1}^N x_{ij} \leq t_i \leq w_{2j} \sum_{i=1}^N \sum_{j=1}^N x_{ij} \quad \forall i, j = 1, 2, \dots, N \quad (12)$$

6 Algorithm pseudocode

- Step A Provide the port of departure (p_d), the port of arrival (p_a), the search area, the three ports connectivity parameter, the sweep angle, the wait times for the vessels at the port (t_w), vessel speed (v_s) and tonnage of each vessel (q_k). The initial optimum route matrix is the one having as its data the departure and arrival ports.
- Step B Loading of all ports of the search area in a $p[1 \dots n]$ matrix, with the exception of the ports that belong in the optimum route matrix. Random selection of port p_i (FOR $i = 1$ to n).
- Step C Check that the connectivity value d_i of this port (p_i) with the departure (p_d) and arrival (p_a) ports, is lower than, or equal to the connectivity parameter defined d_s (that is whether condition $d_i \leq d_s$ applies). If it does not apply, go to Step B and proceed with the next port of the search area, while if the condition applies, go to the next Step D.
- Step D Check whether the intermediate port p_i belongs to the sweep or search area created according to the sweep angle defined. If this is not the case, check first whether the participation of node p_i improves the route generated so far, on the basis of the metric function of the cost for the connection of a node. If it does improve it, integrate the node in the route and continue with Step E. Otherwise, include the node in the prohibition list and go to Step B to continue with the next port.
- Step E Check the stop conditions for all for all optimum routes generated. If they do not apply, continue search in the regions – neighbourhoods of optimum solutions for further optimisation by means of going to Step B, otherwise go to Step F.
- Step F Calculate the total costs for the optimum routes generated, sort end lists of optimum routes on the basis of the cost for each destination.
- Step G Print the final optimum routes for each destination.

7 The flowchart (Figure 6) of the algorithm searching optimum routers

Figure 6 Algorithm flowchart



8 Analysis of the algorithm and documentation of its accuracy

In this section, we shall analyse the algorithm and demonstrate that the search process always results in optimum solutions. Assume graph $G = (V, E)$ where V is the total number of ports in the search area and E is the total number of possible connections. We define the following:

N the total number of nodes (ports) and V the totality of all distances between the nodes

L the total number of acceptable nodes (the ports included in the optimum route)

C the total number of candidate nodes (ports)

$P_d = s$ the port of departure

$P_a = t$ the port of arrival

d_{ij} the distance between two nodes (ports) i, j

p_c the connectivity parameter of two nodes (ports)

φ_c the sweep angle.

The following apply:

```

dij = ∞ if (i,j) ∉ E
L = {s, t} and C = V-L
ΣDj = dsj for j > s and ΣDt = dst is the shortest path s -> t
array_opt {} = EmptySet /*Optimum route
do while /*Loop j ∈ V
for all i ∈ V
    if pi <= pc /* if connectivity condition applies for each node j
        if φi <= φc /* if node j belongs to the sweep area
            ΣDi = min ΣDj (j ∉ L)
            C=C - {i}
            L = L U {i}
            ΣDj := min [ΣDj, ΣDi + dij], j ∈ C
            array_opt = add(i) /* add node to route
        endif
    endif
next
enddo

```

8.1 Analysis

For each node $u \in L$, distance $d(u)$ is the shortest path $s \rightarrow u$:

- a for $n = 1$, $|L| = 1$ applies $d_s = 0$
- b for $n = 2$, $|L| = 2$ applies $d_{st} = \text{Distance}(s, t)$
- c inductive for $n = k$ applies, $|L| = k \geq 1$.

The shortest path $s \rightarrow u$ (Figure 7) is $\Sigma D(u)$.

Addition of a node, v , to L (Figure 8).

Applies $\Sigma D(v) = \Sigma D(u) + d(u, v)$.

Assume that we have another shortest path P' : $s \rightarrow v$ and the preceding node that does not belong to L is node w .

y is the node before w and $\Sigma D(y)$ is the shortest path $s \rightarrow y$ (Figure 9).

Figure 7 The shortest path $s - u: \Sigma D(u)$

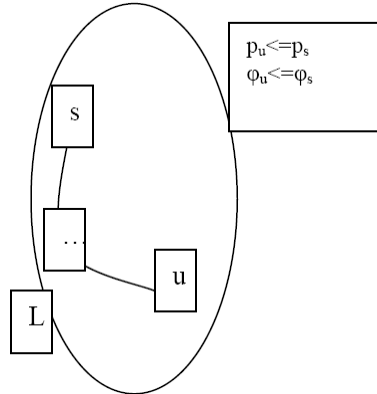


Figure 8 The shortest path $s - v$

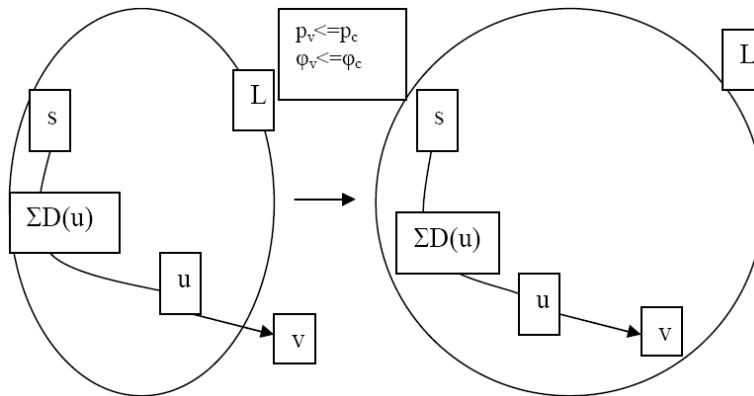
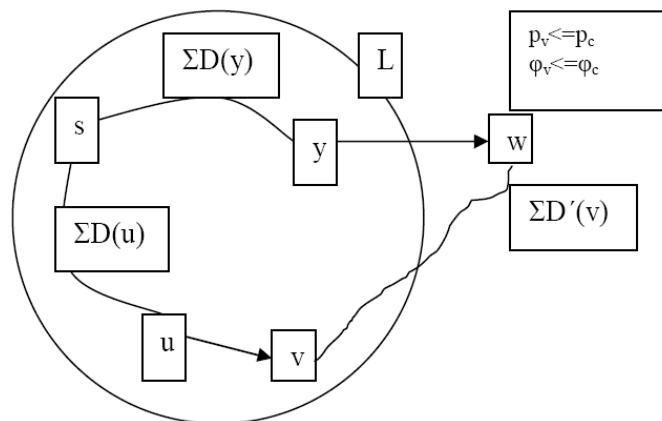


Figure 9 Shortest path $s \rightarrow v$



The following apply, consecutively:

$$\Sigma D'(v) = \Sigma D(y) + d(y, w) + d(w, v) \geq \Sigma D(y) + d(y, w) \geq \Sigma D(w) \geq \Sigma D(v)$$

Indeed, $\Sigma D'(v) = \Sigma D(y) + d(y, w) + d(w, v)$ obviously applies.

Moreover, it applies that: $\Sigma D(y) + d(y, w) \leq \Sigma D(y) + d(y, w) + d(w, v)$ if we deduct the positive term or zero $d(w, v)$.

However, given that node w is outside L , it applies: $\Sigma D(w) \leq \Sigma D(y) + d(y, w)$.

Thus, $\Sigma D'(v) \geq \Sigma D(v)$.

The algorithm always terminates. In each case we have a solution, direct connection $s \rightarrow t$ which is the shortest path between s and t .

9 Optimum routes

The algorithm was tested on the region of the Aegean Sea, having as a cost criterion the total distance travelled and its results were compared with respective coastal routes found in the related bibliography (Aifantopoulou, 2004) for 50 ports. Tests were performed on a PC with CPU Xeon 3.4 GHz with 4 GB RAM. The maximum time for the generation of optimum routes for each destination, did not exceed 4 minutes. The comparison results are presented in the following Table 1, in which the search algorithm presented in this essay showed the best results for the majority of the routes. In the first part of Table 1, there are the optimal itineraries generated by our heuristic algorithm, with the total distance of each itinerary. In the second part, there are the itineraries of the genetic Aifantopoulou algorithm with the total distance of each itinerary. In addition, the total covered distance of the SETHAM survey is also mentioned for the same limitations. The compared results show that, the total covered distance of the optimal itineraries of Aifantopoulou and SETHAM algorithms for the same destinations, is less by 45% and 49%, respectively.

Table 1 Comparison of routes with genetic algorithm for 50 ports

<i>Code</i>	<i>Routers heuristic algorithm</i>	<i>Miles</i>
Opt.1	PIRAEUS-AGIOS KIRIKOS IKARIAS-ΦΟΥΡΝΟΙ-SAMOS KARLOVASI-SAMOS VATHI	183
Opt.2	PIRAEUS-SERIFOS-SIFNOS-FOLEGANDROS-SIKINOS-ΙΟΣ-THIRA	144
Opt.3	PIRAEUS-PSARA-XΙΟΣ-MITILINI	217
Opt.4	PIRAEUS-PATMOS-LIPSI-KALIMNOS-KOS-SIMI-RODOS	286
Opt.5	PIRAEUS-SIROS-PAROS-NAXOS-KOUFONISI-SXINOUSA- IRAKLIAIOS-SIKINOS-FOLEGANDROS-THIRA	190
Opt.6	PIRAEUS-RETHIMNO-XANIA-KISSAMOS-AGIA PELAGIA	282
Opt.7	PIRAEUS-SIROS-PATMOS-LEROS-KALIMNOS-KOS- CHALKIRODOS	296
Opt.8	PIRAEUS-ΙΟΣ-THIRA-ANAFI	153
Opt.9	PIRAEUS-PAROS-NAXOS-DONOUSA-AMORGOS AIGIALIASTIPALAIΑ	185
Opt.10	PIRAEUS-ANDROS-AGIOS EYSTRATIOS-LIMNOS MIRINA	190
Opt.11	PIRAEUS-AGIOS NIKOLAOS-SITIA-KASOS-DIAFANIO KARPATHOY	300

Table 1 Comparison of routes with genetic algorithm for 50 ports (continued)

<i>Code</i>	<i>Routers heuristic algorithm</i>	<i>Miles</i>
Opt.12	PIRAEUS-XANIA	147
Opt.13	PIRAEUS-KITHNOS-SERIFOS-SIFNOS-KIMOLOS-MELOS	109
Opt.14	PIRAEUS-SIROS-TINOS-MIKONOS-MIRINA-THESSALONIKI	381
Opt.15	PIRAEUS-FOLEGANDROS-THIRA-KATAPOLA	165
Opt.16	PIRAEUS-SIROS-TINOS-MYKONOΣ	99
Opt.17	PIRAEUS-HPAKAEIO	173
<i>Code</i>	<i>Routers Aifantopoulou</i>	<i>Miles</i>
R1	PIRAEUS-DONOUSΑ-AGIOS KIRIKOS IKARIAΣ-SAMOS VATHISAMOS KARLOVASI	207
R2	PIRAEUS-MYKONOΣ-KIMOLOS-FOLEGANDROS-SIFNOS	195
R3	PIRAEUS-MITILINI-XIOΣ	239
R4	PIRAEUS-PATMOS-KALIMNOS-KOS-SIMI	261
R5	PIRAEUS-KOUFONISI-IRAKLIA-AMORGOS KATAPOLAASTIPALAIA-LEROS-FOURNOI	269
R6	PIRAEUS-XANIA-RETHIMNO-KIΣSAMOS-AGIA PELAGIA	295
R7	PIRAEUS-SITIA-KASOS-DIAFANIO KARPATHOY-RODOS	350
R8	PIRAEUS-THIRA-IOS-ANAFI	178
R9	PIRAEUS-PAROS-NAXOS-SXINOUSA-AIGIALI	147
R10	PIRAEUS-TINOS-SIROS-KITHNOS-ANDROS-MIRINA	287
R11	PIRAEUS-IRAKLIO-MELOS-KITHNOS-KOS-CHALKI-RODOS	546
R12	PIRAEUS-MIKONOS-PAROS-DIAFANIO KARPATHOY-AGIOS NIKOLAOS	353
R13	PIRAEUS-XANIA-RETHIMNO-IRAKLIO-FOLEGANDROS-SIKINOS	303
R14	PIRAEUS-TINOS-SIROS-SERIFOS-SIFNOS-SAMOS VATHI-SAMOS KARLOVASI	277
R15	PIRAEUS-IOΣ-NAXOΣ-MELOS-THIRA	249
R16	PIRAEUS-THIRA-KATAPOLA-PATMOS-LIPSI-KALIMNOS	252
R17	PIRAEUS-PAROS-MIKONOS-MIRINA-THESSALONIKI	398
R18	PIRAEUS-MITILINI-XIOS-PSARA	281
	Total distance heuristic algorithm for 50 ports	3.514
	Total distance algorithm Aifantopoulou for 50 ports	5.087
	Total distance SETHAM Project for 50 ports	5.252
	Difference with Aifantopoulou	-1.573
	Defference with SETHAM	-1.738
	Optimisation with Aifantopoulou	-45%
	Optimisation with SETHAM	-49%

10 Conclusions

The algorithm presented in this essay approached routing problems with capacity limitations, as well as with spatial and time limitations, in sea regions presenting with specific particularities. The case examined could be referred to as vessel routing problem with capacity, time, and space limitations (VRP with CTPC). The algorithm can be applied to any sea region both with smooth and irregular island distribution, as it can be adapted to the particularities of each area. Moreover, the algorithm can be exploited in generic vessel routing problems, when we are invited to formulate the search area using geographic and topological limitations.

Moreover, the algorithm can be used in general problems of network design, of any form, in particular when we have parameters related to the structure and topology of the network application area. Moreover, it can be used in routing problems in which routes are dynamic, such as the dial-a-ride problems.

The algorithm can also be used for searching optimum solutions in critical problems that concern a maritime company, such as the issues, of the sea regions in which they should operate and what kind of vessels they need in such regions. In combination with the above, an objective of each maritime company, as well as for maritime organisations is to design and implement an integrated decision support system for the coastal transports issue. Such a dynamic system can constitute a strategic tool for a maritime company, with the objective to establish adequate strategies and policies with regard to the most effective lines for routing its vessels, as well as for the formulation of an appropriate fleet for covering market needs.

References

- Aifantopoulou, G. (2004) *Implementation of Experienced Technicians and Systems Optimum Design of Vehicle Fleet Distribution Routes*, PhD dissertation, School of Civil Engineering, Aristotle University of Thessaloniki, Greece.
- Christofides, N., Mingozzi, A. and Toth, P. (1979) 'The vehicle routing problem', *Combinatorial Optimization*, pp.313–338, John Wiley & Sons, Chichester.
- Cordeau, J-F., Laporte, G. and Mercier, A. (2001) 'A unified tabu search heuristic for vehicle routing problems with time windows', *Journal of the Operational Research Society*, Vol. 52, No. 8, pp.928–936.
- Desrochers, M., Desrosiers, J. and Solomon, M.M. (1992) 'A new optimization algorithm for vehicle routing problems with time windows', *Operations Research*, Vol. 40, No. 2, pp.342–354.
- Laporte, G. and Osman, I.H. (1995) 'Routing problems: a bibliography', *Annals of Operations Researcher*, Vol. 1, No. 61, pp.227–262.
- Louis, J.S., Yin, X. and Yan, Z.Y. (1999) 'Multiple vehicle routing with time windows using genetic algorithms', *Proceedings of the 1999 Congress on Evolutionary Computation*, Department of Computer Science, University of Nevada, Washington DC, IEEE Press, pp.1804–1808.
- Savelsbergh, M.W.P. (1985) 'Local search for routing problems with time windows', *Ann. of Operations Research*, Vol. 4, No. 6, pp.285–305.
- SETHAM Project-Ministry of Merchant Marine of Greece (2001) 'Development of a Management System for National Maritime Transport in Greece', Final Project Report of a Study Conducted by TRUTH S.A., Trademco, Alpha Marine, NEI/MERC, Athens.

- Solomon, M.M. (1987) 'Algorithms for the vehicle routing and scheduling problem with time window constraints', *Operations Research*, Vol. 35, No. 6, pp.254–265.
- Tang, H. and Miller-Hooks, E. (2005) 'A tabu search heuristic for the team orienteering problem', *Computer Operations Research*, Vol. 32, No. 6, pp.1379–1407.
- Tang, J., Kong, Y., Lau, H. and Ip, A.W.H. (2010) 'A note on efficient feasibility testing for dial-a-ride problems', *Operations Research Letters*, Vol. 38, No. 10, pp.405–407.
- Toth, P. and Vigo, D. (2002) *The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications*, SIAM, Philadelphia, PA.
- Vidal, T., Gräimic, T.G., Gendreau, M. and Prins, C. (2011) *A Unifying View on Timing Problems and Algorithms*, CIRRELT Working Paper (43).