Near-Optimal Covering Solution for USV Coastal Monitoring using PAES

Hand Ouelmokhtar, Yahia Benmoussa, Jean-Philippe Diguet, Djamel Benazzouz, Laurent Lemarchand

Near-Optimal Covering Solution for USV Coastal Monitoring using PAES

Hand OUELMOKHTAR[1*], Yahia BENMOUSSA[1†], Jean-Philippe Diguet[2†], Djamel Benazzouz[1†] and Laurent Lemarchand[3†]

[1*]LMSS, University M'Hamed Bougara, Boumerdes, Algeria.
[2]Crossing/CNRS, CNRS, Australia.
[2]Lab-STIC, UBO, France.

*Corresponding author(s). E-mail(s):
h.ouelmokhtar@univ-boumerdes.dz;
Contributing authors: y.benmoussa@univ-boumerdes.dz;
jean-philippe.diguet@cnrs.fr; d.benazzouz@univ-boumerdes.dz;
laurent.lemarchand@univ-brest.fr;
†These authors contributed equally to this work.

Abstract

This paper addresses a multi-objective optimization problem for marine monitoring using USV. The objectives are to cover the maximum area with the lowest energy cost while avoiding collisions. The problem is solved using an exact and heuristic methods. First, a multi-objective Mixed Integer Programming formulation is proposed to model the USV monitoring problem. It consists of a combination of the Covering Salesman Problem (CSP) and Travelling Salesman Problem with Profit (TSPP). Then, we use CPLEX software to provide exact solutions. On the other hand, a customized chromosome-size algorithm is used to find heuristic solution. The latter is a multi-objective evolutionary algorithm known as Pareto Archived Evolution Strategy (PAES). The obtained results showed that the exact solving of the USV monitoring mission problem with mixed-integer programming (MIP) methods needs extensive computational costs. However, the customized PAES was able to provide Near-optimal solutions for large-size graphs in much faster time as compared to the exact one.

1 Introduction

Nowadays, unmanned Surface Vehicles (USVs) are more and more used in both military and civil fields to replace humans in performing ocean and marine tasks, especially in hostile and dangerous environments [1, 2]. The main advantages of USVs are that they do not have restrictions imposed by a human crew member, such as temperature, space, or environment disturbances, as well as their ability to evolve in environments where humans are not able to intervene safely, in addition to their cost and continuous activity.

There is several applications of USVs such as environmental monitoring [3], ocean resource exploration [4], maritime search and rescue [5], ocean weather forecasting [6] and defense.

In this paper, we address the general problem of marine monitoring using an USV equipped with on-board LiDAR (Light Detection and Ranging) allowing the remote covering of distant points. In fact, optical sensors such as LiDAR and cameras have been successfully used for many robotic applications, with the aim of providing information about the navigation environment located within their visibility range [7] For example, Fig- 1 represents the USV monitoring mission of Brest city port, in which the USV moves from the starting point to visit all the selected way-points (green point) then comes back to the departure point. The red lines represent the USV tour where all the points located within the LiDAR range are considered as covered. The objectives assigned to the USV mission are first to maximize the covered area and second minimize the consumed energy (distance covered). To reflect real operating conditions, we consider that the USV operates in an environment with tidal current effects and should ensure obstacle avoidance along its mission.

Basically, this kind of problems are modeled using the well known Covering Salesman Problem (CSP) which is the generalisation if the Traveling Salesman Problem (TSP). For example, several CSP variants have been proposed in [8–12], and most of them are published in TSPLIB [13], data base which stores a set of pre-solved use cases of (TSP, CSP, TSPP, etc.) and used as reference solutions. However, these studies do not consider LiDAR-based coverage which is more realistic than standard CSPs. Moreover, they are exclusively mono-objective and are not able to deal with multiple conflicting objectives.

To solve USV the above described mission monitoring problem, two approaches are proposed : The first one is an exact solution based on Mixed Integer Programming (MIP) modeling [14] of the combination of two well-known operation research problem formulations : Traveling Salesman Problem with Profit (TSPP) [15] and Covering Salesman Problem (CSP) [8]. Actually, LiDAR coverage is modeled using the concept of CSP customers. Then, the CSP is used in a non-Hamiltonian TSPP tour with two conflicting objectives. The MIP model is then solved using CPLEX optimizer tool to provide the optimal USV tour. We found that while the solution found is optimal, the computational time is extremely high.

The second one is a heuristic method based Pareto Archived Evolution Strategy (PAES) [16] which is a multi-objective algorithm inspired from genetic algorithms (GA). First, the adaptation of the standard PAES to the USV covering problem is called S-PAES. Then, in order to enhance the performance and the solution quality,

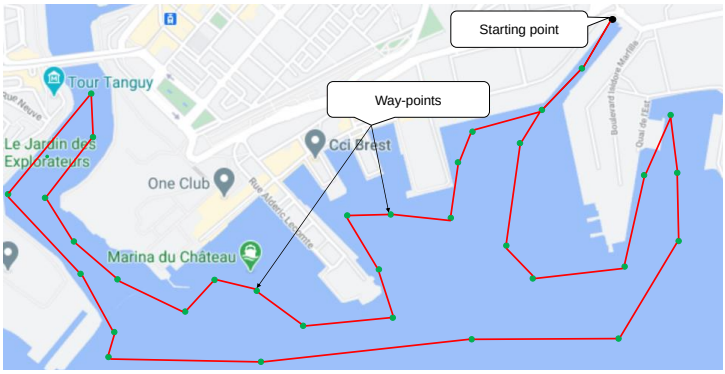

Fig. 1 Brest city harbor monitoring mission

a customized PAES (C-PAES) is proposed. It improves the efficiency of the direct approach of S-PAES by the calculation of an optimal chromosome size allowing avoiding way-points covering redundancy.

Then, the exact and heuristic solutions are compared. The obtained results show that, for small size problems (graph with total vertex equal to or less than 74 vertices), C-PAES is able to provide an optimal solution in a much shorter time than the exact method. Beyond that size, the exact solver is not able to provide a solution in an acceptable time, whereas C-PAES can converge to satisfactory solutions for relatively large graph sizes. As example, when solving the monitoring problem with a graph of 2596 vertices, the obtained solution for the entire coverage area has only 7% additional energy consumption as compared to that of a 74-vertex graph. However, the computational time required by PAES to solve the 2596-vertex problem (120 second) is about 400 time less than the computational time required to solve the 74-vertex problem with the exact approach (51333 second).

The rest of the paper is organized as follows: Section 2 lists the most relevant related work. Section 3 describes the developed methodology in this study. The problem formulation is described in detail in Section 4. The problem modeling and solving is given in Section 5. Solver's implementations are described in section 6. The obtained simulation results are analyzed and discussed in Section 7. A conclusion and some perspective works are presented in Section 8.

2 Related Work

2.1 Covering Salesman Problem

The first CSP formulation was developed by Current and Schilling [17] in 1989. to solve, they proposed a two step composed heuristic algorithm. First, a set of vertices (SCP) that allows to cover all the vertices is defined. then, the optimal TSP tour over the pre-defined SCP should be calculated in the second step. [8] proposed two local search algorithms denoted named LS1 and LS2. LS1 applies removal and reassignment of moves where some cities are removed from the tour and other ones will be reinserted to obtain a new optimal solution. LS2 uses suppression-reassignment

as well as perturbation procedures to improve the solution quality. It employs the suppression-reassignment heuristic to find a subset of cities, and then applies the Lin-Kernighan heuristic proposed by [18] on this subset to find a TSP tour passing through its constituent cities. [9] provide integer linear programming (ILP) based solution for the CSP. Starting from an initial feasible solution, the algorithm applies the destroy-repair paradigm to optimize the tour length. For this purpose, some vertices are removed and reallocated forming a new feasible solution resulting from solving an ILP-based model to optimally. On 2015, [10] developed an hybrid ant colony optimization (ACO) algorithm with dynamic programming to solve the CSP problem. On the other hand, this study provides also a MIP formulation which is further solved with CPLEX for the small instances. [11] developed two hybrid metaheuristic for CSP solving which are: Artificial Bee Colony algorithm (ABC) and Genetic Algorithm (GA) with using new local search based operators. In addition to specific knowledge during crossover and mutation in GA. The comparison shows that GA is able to improve the best known solutions for most large instances.

Most CSP works consider single objective that is the minimization of the travel cost required to cover all vertices. [12] developed a new CSP formulation called 'A multi-objective Covering Salesman Problem with k-coverage'. The proposed CSP considers two conflicting objectives that are tour length minimization and coverage rate maximization. the prefix k consists on the number of nodes that can be covered by one vertex. The problem is solved using the Non-dominated Sorting Genetic Algorithm (NSGA-II). Computational experiments are performed on ten numerical examples taken from TSPLIB by setting k=2. Authors confirm that the higher is the k-coverage value, higher is the robustness of the system. However, the tour cost (tour length) increases, and beyond a certain value of k, the tour cost becomes unrealistic.

2.2 USV covering path planning

Most of existing studies about USVs mission planning focus on path planning [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. However, the objective in these works is to find an optimal route from a starting point to an arrival point which is not suitable for monitoring mission where the main objective is to cover the maximum navigation area.

There are only few studies which consider on the covering tour problem for USVs. For example, in [30], authors developed a mission planning for USV when operating in a non-obstructed sea environment. The proposed solution allows to visit a set of points in a limited time starting from and returning back to the same points. Each point is associated with a profit named mission profit and the Euclidean inter-points distances are calculated and transformed into time cost at the beginning by assuming a constant USV speed. The problem is given by the Traveling salesman Problem (TSP) without considering the constraint of visiting all nodes, as it causes a conflict with the limited operating time assumed in their study. The authors proposed a Genetic algorithm to solve this problem where the objective function aims to optimise the ratio of the sum of performed missions profit and the sum of all profits under constraint of limited operating time.

[31] proposed a solution for covering reservoir in order to monitor water quality by collecting samples. The problem is modeled as a TSPP (TSP with profits), in which a minimum number of nodes must be visited by assuming that each nodes represents a certain coverage of water area. Actually, TSPP [15] is a bi-objective TSP where it is not necessary to visit all vertices. Considering that a profit is associated to each vertex, the objective is to monitor a maximum of the coverage area in one tour under constraints of limited operating time and maximum volume of water samples that can be loaded by the USV.

2.3 Scope of the study

These mentioned studies did not consider the covering objective in the context of monitoring mission, in which the vertices to be visited are predefined and associated with an abstract and fixed profit to be maximized. However, in case of a monitoring mission, the USV can select any vertex while achieving the mission. Moreover, the covering objective dynamically depends on different parameters such as the maps topography (borders, obstacles, ... etc) and LiDAR range.

In this paper, we considered the USV monitoring mission as a mix of CSP and TSPP, which has not been considered yet, as far as we know for the CSP part [17], the idea is to find the lowest cost Hamiltonian tour by visiting a subset of vertices (facilities) that allow to cover the non-visited one (customers) included in the predetermined coverage distance. The selected vertex to visit should be a facility and the other ones included in LiDAR coverage rate will be the customers. Moreover, we propose a combination of the mono-objective CSP with the TSPP which leads to a generation of new problem formulation that makes the concept of the CSP to be considered as a profit in the bi-objective non-Hamiltonian TSPP. This leads to a multi-objective optimization problem, which will be solved by exact and heuristic approaches. The exact method consists in solving the generated MIP model of the bi-objective CSP using CPLEX optimization tool. The proposed heuristic method is a multi-objective evolutionary algorithm called Pareto archived evolution strategy (PAES) [32]. The exact method will be used as a reference in the benchmark.

3 Methodology

In this section, we present our methodology which is subdivided into three phases as described in Fig- 2

3.1 Problem formulation

The objective of this phase is to provide formal descriptions of the environment where the USV is evolving including : Navigation area, LiDAR, tidal currents and consumed energy as shown in Fig- 2.a. All these formulations are used by both exact and heuristic solutions as described hereafter. The area to monitor is described as a grid of points (vertices) and edges associated to movements between neighbour nodes. In this study, the USV speed is considered as constant for both exact model and heuristic approach. Thus, the arrival time from one point to another is not controlled.

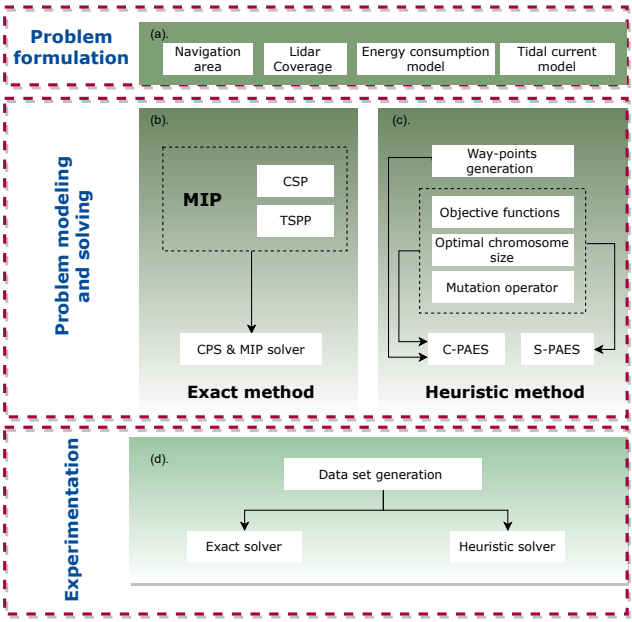

Fig. 2 Methodology

3.2 Problem modeling and solving

The USV covering tour problem is modeled and solved using two methods exact and heuristic methods.

3.2.1 Exact method

The problem is modeled as a combination of CSP and TSPP. Actually, CSP is used to model the LiDAR coverage using the concept of CSP customers. To consider the energy and covering conflicting objectives, CSP is integrated into a non-Hamiltonian TSPP. In fact, TSPP is a generalization of TSP considering a profit when visiting a given vertex. In our case, the profit serves to represent the coverage rate.

Then, as shown in Fig- 2.b the problem is formalized using MIP and solved using a MIP solver. We use the cutting plane scheme adapted from TSP techniques for eliminating sub-tours during MIP solving [33]. The objective is to find rapidly a primary solution, which may not fulfill all constraints. Then, the remaining constraints are integrated in a second step to refine the founded solution [34].

3.2.2 Heuristic method

In the heuristic method, we use the PAES algorithm. As shown in Fig- 2.c, the algorithm begins by generating initial solutions as sequences of way-points that define the complete trajectories, in which each two successive way-points are connected by the lowest energy cost path. After that, it will be evaluated according to the energy and

covering objective functions. Therefore, their fitness values will be enhanced using the mutation operator with four modes.

It is obvious that the larger is the number of vertices representing the environment to be monitored, the more precise is the trajectory. Conversely, the larger the number of vertices, the greater the complexity of the problem, which affects the computation time. For this purpose, we have implemented a methodology to enhance the above cited standard PAES (S-PAES). The idea is to calculate the optimal chromosome size parameter by exploiting the fact that the vertices covered by the LiDAR do not need to be visited, which allows to enhance both solution quality and computation time. We note this customized PAES as C-PAES.

3.3 Experimentation

In this phase, we generate, first, the data set which are all the global configuration input parameters including navigation area, LiDAR coverage and tidal currents. They are shared between both exact and heuristic solvers and should be provided to them according to the predefined format. Secondly, two solvers are implemented for both exact and heuristic methods to process the generated data set and to provide the solutions (See Fig- 2.d).

4 Problem formulation

4.1 Navigation area

The targeted area is represented by the binary grid map using black and white colors that indicate obstacles and free spaces respectively. The map is used to produce a graph $G(V, A)$ where, V is the set of vertices and A is the set of arcs. The white area is divided into a grid of cells where each cell represents one vertex v_i. The adjacent vertices considering cardinal and diagonal neighbourhood will be connected by weighted arcs $e \in A$ such that the distance is considered as the arc's weight w_e.

4.2 LiDAR covered points

In order to determine the LiDAR coverage area for the USV location, a list of viewed-points C_i is associated to each vertex $v_i \in V$ depending on the LiDAR sensor radius. This is shown in Fig- 3, where the viewed points from the visited vertex v_i are the green points pertaining to the LiDAR coverage radius represented by red circle. On the other hand, the blue points are the vertices obstructed by an obstacle. Finally, the black points are those outside of the LiDAR coverage radius.

4.3 Tidal Current model

Without loss of generality but for simplicity, we suppose that the tidal currents vary under sinusoidal function, over a period of 12 hours. We have also considered that the currents vary only as function of time which means that at a given time t, the currents have the same intensity and direction over the navigation surface. Figure 4 shows the

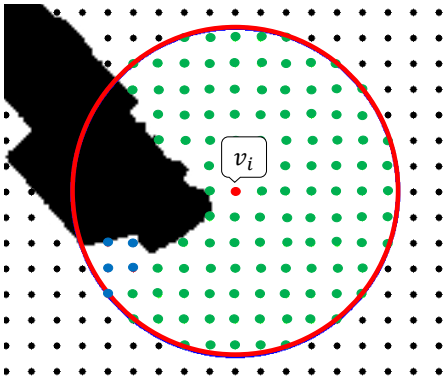

Fig. 3 Viewed points under LiDAR coverage and obstacles intersection

current variation function from the low to the high tide. Eq. 1 gives the current speed variation on the Ox axis.

$$V_{cx}(t) = V_{\max} \sin(\omega t) \tag{1}$$

Where V_{max} is the maximum current speed and ω is the pulsation over a period of 12 hours.

If we consider α the angle between USV displacement and Ox axis, then the current speed is given by Eq. 2.

$$V_c(t) = V_{cx} \sin(\alpha) \tag{2}$$

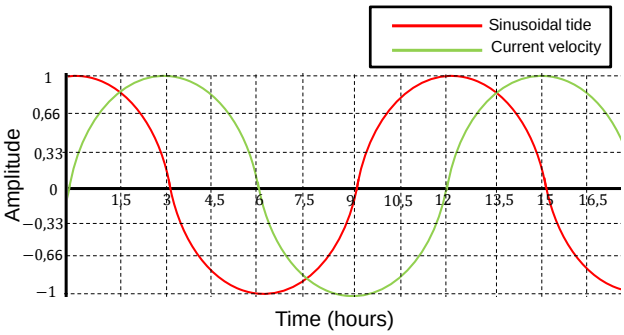

Fig. 4 Sinusoidal tide and current velocity variation

4.4 Energy consumption model

Basically, the USV's total consumed energy is the sum of elementary consumption associated to a moving between two vertices. $\omega_{i,j}$ is the distance between vertices

v_i and v_j, while V_{usv} is the relative USV speed. The required travel time $t_{i,j}$ is calculated according to Eq. 3 .

$$t_{i,j} = \frac{\omega_{i,j}}{V_{usv}} \tag{3}$$

If we consider that V_c is the tidal current speed, then the actual USV speed V_a is given by Eq. 4 .

$$V_a = V_{usv} + V_c \tag{4}$$

Referring to the energy model given by [24], the USV energy expression considering the linear hydrodynamic drag, ship speed and travel time is given by Eq. 5.

$$E_{i,j} = \beta {V_{usv}}^3 \frac{\omega_{i,j}}{V_a} \tag{5}$$

Where $E_{i,j}$ is the energy consumed by the USV when moving from vertex V_i to vertex V_j, β is a constant value that combines water density value, reference area, and drag coefficient. Speed square times β represent hydrodynamic drag. Assuming that the consumed energy is null at the starting location, then Eq. 6 is used to update the consumption when moving from point v_i to point V_j.

$$E_j = E_i + E_{i,j} \tag{6}$$

Where E_i is the total energy consumed to reach the vertex v_i, and E_j is the consumed energy to reach the selected vertex v_j.

5 Problem modeling and solving

In this section, we describe the exact and heuristic methods for the covering problem.

5.1 Exact method

In the exact method, the MIP model is presented based on the combination of CSP and TSPP formulations. The resolution is based on the CPS method combined with a MIP solver.

5.1.1 CSP and TSPP combination

The covering salesman problem (CSP) is a generalization of the TSP in which we have to construct a minimum length tour (minimum energy) satisfying all the customers' demand by visiting or covering them [17]. As described in Fig- 5.a, for each facility (way-point) it is given a covering radius (USV LiDAR range) within all the located customers (vertices pertaining to the monitored area) that will be covered.
In CSP, the solution should achieve the complete converting of all the vertices. To transform the coverage rate to an additional objective, we use the concept of profit introduced in TSPP. Actually, in TSPP, as given in Fig- 5.b in TSPP, it is not necessary to visit all vertices. A profit is associated with each vertex. The overall goal is to collect a maximum of profit with minimum travel costs.

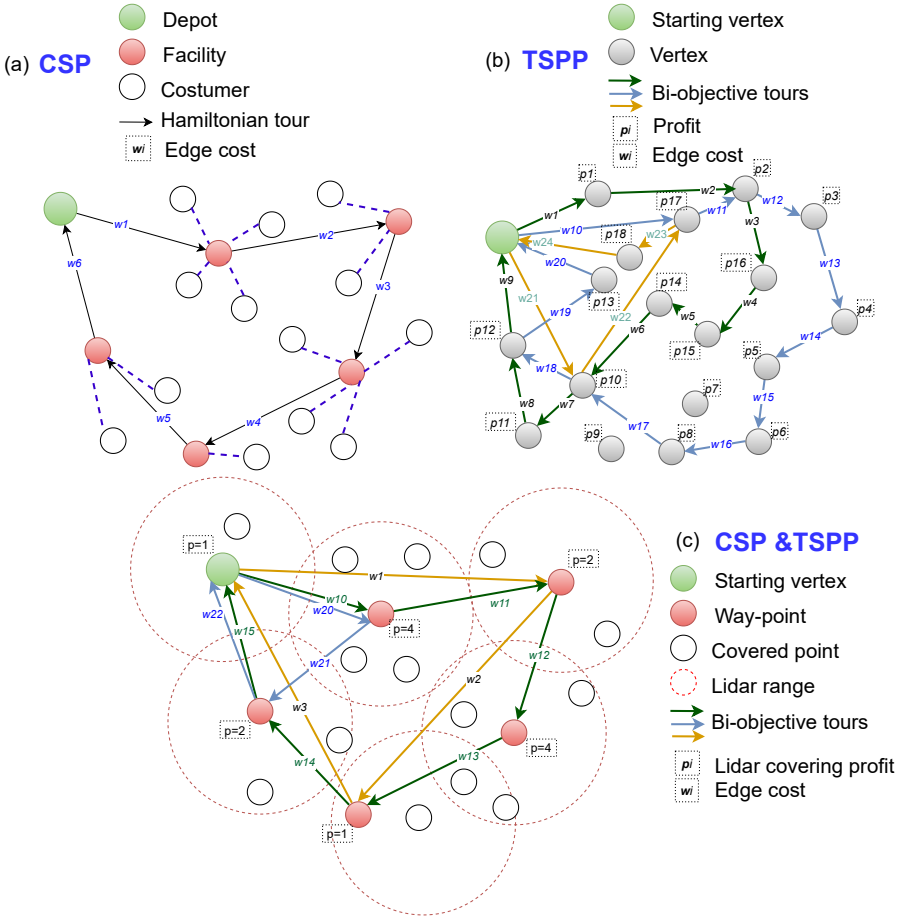

Fig. 5 Combination of CSP and TSPP

The combination of CSP and TSPP allows us to model the coverage problem by considering both energy and coverage objectives. Figure- 5.c shows the sum of vertices covered by the LiDAR range (customers) which represents the TSP profits associated to each visited way-point (facility).

5.1.2 MIP modeling

In this section, we will use the MIP to model the above described CSP and TSPP formulations. For the given graph $G(V, A)$, where V and A represent the graph's vertices and their edges sets respectively as determined in section 4.1. $\forall e \in A, w_e$ is the distance cost of the arc e. Let $v_1 \in V$ be the starting vertex, according to the LiDAR visibility, we assume that $\forall v \in V, \gamma(v)$ is the set of vertices $v' \in V$ allowing to cover the vertex v. The developed MIP model is given according to the following

equations:

$$\min \sum_{e \in A} \beta {V_{usv}}^3 \frac{\omega_e}{V_a} x_e \tag{7}$$

$$\min n - \sum_{v \in V} z_v \tag{8}$$

Subject to:

$$\sum_{e \in \delta(v)} x_e = 2y_v, \forall v \in V, v \neq v_1 \tag{9}$$

$$\sum_{e \in \delta(S)} x_e \geq 2y_v, \forall v \in S \subset V, S \neq \emptyset, v_1 \notin S \tag{10}$$

$$z_v \leq \sum_{v' \in \gamma(v)} y_{v'}, \forall v \in V \tag{11}$$

$$x_e, y_v, z_v \in \{0, 1\}, \forall v \in V, \forall e \in E \tag{12}$$

Such that, x_e, y_v, and z_v are the binary decision variables (Eq.(12)). x_e indicates if the edge e is a part of the tour or not. y_v and z_v represent the visited and covered states of the given vertex v respectively.

Equation (7) represents the first objective function which minimize the energy costs (subsection 4.4). The profit maximization objective is performed according to Eq. 8 by minimizing the total of non-covered vertices.

Like the classical TSP, Ed.(9) ensures that each visited point is connected to two edges except the starting vertex. However, this is valid only in case the point is visited, i.e. those for which $y_v = 1$. Equation (10) is adapted from the one that allows avoiding sub-tours in TSP (with notation $\delta(S \subset V) = e, e = (v - v'), v \in S, v' \notin S$. For each sub-set S of vertices, we need an edge e to come to the set, and another one to leave it. Unlike TSP, this is true if and only if one vertex of the subset is visited. Finally, Eq.(11) allows to check the covered points by the visited ones. With this formulation, the problem associated to a graph $G = (V, E)$ with a LiDAR range allowing a point to cover on average ω neighbours contents $3.\mathsf{V} + \mathsf{A}$ decision variables, and $(\mathsf{V} - 1) + (\omega.\mathsf{V}) + (2^{\mathsf{V}-1})$ constraints (for equation families (9), (11), and (10)). Obviously, sub-tours elimination constraints can not be generated directly for large graphs.

5.1.3 MIP solving

The developed MIP model is solved with MIP solver combined with Cutting Plane method. The two objectives are aggregated in a single objective function with the weights α and $(1 - \alpha)$ where $\alpha \in [0;\ 1]$. When solving the MIP, if any sub-tour appears, the corresponding sub-tour elimination constraint (cutting plane) is added to the MIP model [33], then we restart the calculation until getting a valid optimal solution for this α value. in order to find another optimal non-dominated solution, α will be increased by 0.1, then we restart the calculation until $\alpha = 1$, as described in algorithm. 5.1.3.

Algorithm 1 Plane scheme and MIP solver algorithm

create MIP with:
$obj_1 = eq.(7)$
$obj_2 = eq.(8)$
$Sols = \emptyset$

for $\alpha \leftarrow 0.0$ **to** 1.0 **by** 0.1 **do**
  generate MIP with:
  $obj = \alpha * obj_1 + (1 - \alpha) * obj_2$
  subject to: Eq.(9), (11), (12)
  $solution$ = solve MIP (with CPLEX)
  while *ThereIsSubTour* (S) **do**
  Eliminate S by adding corresponding eq.(10) to MIP
  $solution$ = solve MIP
  add $solution$ to $Sols$
Result: $Sols$

5.2 Heuristic method

The heuristic method consists on specific adaptations of the PAES algorithm for solving the covering problem. As shown in the pseudo-code 5.2,the PAES start with generating only one solution (current solution). then using mutation operator, another mutated solution (m) will be created at each generation. Pareto dominance concept will be also used to examine the new solution m, If m dominate c, c will be updated with m. else the dominance will be examined between m and other ones stored in the PAES archive. Finally, an update of the PAES archive will be done. This mechanism will be repeated until reaching the determined number of generation.

Algorithm 2 PAES algorithm

Set Algorithm parameter
Create current solution (c) randomly
Evaluate and add to archive

for ı ← 1 **to** $max_{generation}$ **do**
  Mutate (c) and generate new candidate (m)
  if c *dominates* m **then**
    $m \leftarrow c$
    Update PAES archive
  else
    Compare (c) with archive member
    Update PAES archive
  Select new current solution
Present results

In order to improve the performance of the PAES algorithm, the methodology for computing the optimal number of way-points is adopted. In this methodology, we introduce an additional parameter N that represents the new size of the chromosome. It is supposed to enhance the PAES solution quality. The new chromosome size corresponds to the maximal number of way-points to visit required to cover the whole existed points. Equation 13 corresponds to the new chromosome size N, where n is the total number of vertex and avr_{Disc} represents the average number of covered points for each vertex while eliminating the redundant ones. To determine avr_{Disc}, an offline scanning of the Map is done, while considering both LiDAR range, Map topology and inter-vertex distance.

$$N = \frac{n}{avr_{Disc}} \tag{13}$$

For further details refer to [16].

6 Experimentation

6.1 Data set generation

To be processed by the exact and heuristic solvers, the considered problem should be transformed to a predefined format. Figure- 6.a illustrates the procedure. Firstly, the binary grid map image where land and sea are represented with black and white colors is generated from a Google-Map image of the area to be monitored. Secondly, given the distance between each two adjacent vertices, the graph $G(V, A)$ (1) is generated where V and A represent the vertices and edges sets respectively. The LiDAR radius parameter (2) serves to set the covered points $\{C\}$ associated to each vertex.

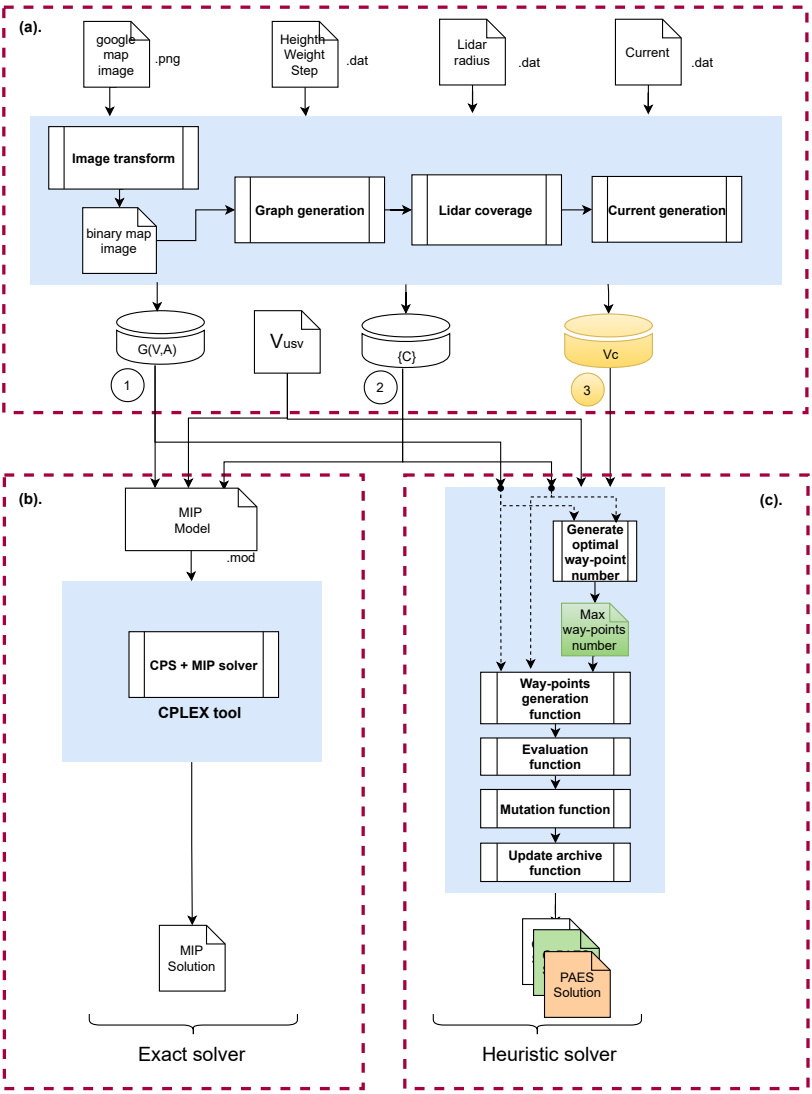

Fig. 6 Exact and heuristic solvers

Finally, (3) the tidal current velocities and orientations specific to each vertex are set for each vertex in V_c.

Here, we propose a simple test environment, illustrated in Fig- 7, on which we will run our solvers. It represents an area with a $2km^2(2 * 1km)$ total surface. The black and white colors represent obstacles and free area respectively. The LiDAR coverage radius (LCR) is set to $200m$ which is representative of the LiDAR capabilities used is real USVs. This test environment is then transformed to G_1, G_2, G_3 and G_4 graphs depending on the chosen inter-vertex distances d listed in Table 1. The

objective is to evaluate the solvers performance when increasing the number of vertices n. In each graph, the first vertex V_1 is set as a departure point where the USV should start and ends its tour.

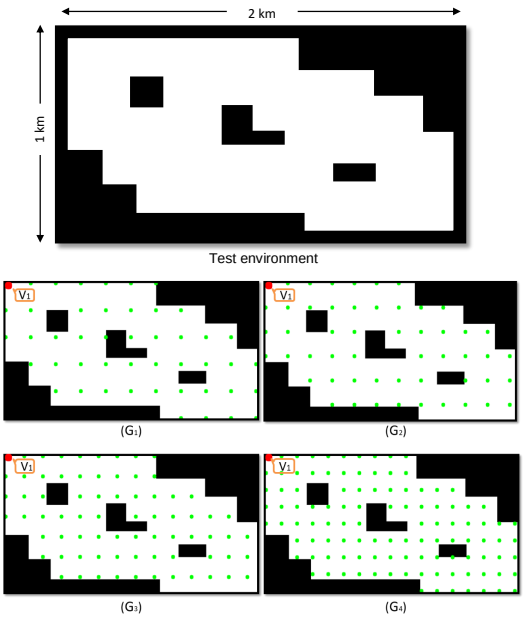

Fig. 7 Different graphs representation for environment test

Table 1 G_1, G_2, G_3 and G_4 graphs data

Graph	G_1	G_2	G_3	G_4
d (m)	200	180	150	125
n	47	53	74	110

6.2 Exact method solver implementation

The exact method's solver is implemented using Optimization Programming Language (OPL) within CPLEX, software optimization tool developed by IBM and integrating exact solvers for various optimisation problem including MIP [35]. Figure- 6.b shows the MIP model defined in .mod file to processes the data set files describing $G(V, A)$ graph, the LiDAR covering sets and the USV speed then appeals the CPLEX optimizer.

6.3 Heuristic method solver implementation

In addition to the above processed parameters, the PAES solver consider the tidal current velocity V_c as described in Fig- 6.c. First, we implemented, using C language, the S-PAES algorithm including the following functions :

- (Random) generation of initial way-point sequences.
- Evaluation of the obtained solution fitness value based on the objective function (energy/coverage).
- Improvement of the solution quality using the mutation operator.
- Update the PAES archive to improve the non-dominated Pareto front.

Thereafter, the C-PAES is implemented by adding the function to compute the maximum number of way-points which is used as an additional parameter to tune the chromosome size.

7 Results and Comparison

In this study, we have described the obtained simulation results in detail. We, firstly, start by comparing between exact and heuristics approaches in terms of performance and solution quality. Secondly, we evaluate the convergence of the proposed PAES solver. Finally, we use the heuristic approach to find the efficient tour to monitor the Brest city harbor. For this study case, we show how PAES solver can be used to select the best departure time to save energy by leveraging tidal currents effects.

The developed programs of simulations are executed on a PC running Linux (ubuntu 16.04) OS and equipped with 4 GB of RAM and quad-core intel i5-3320/1.9 GHz processor.

the table bellow (table.2 represents the values of the constants considered in this study

Table 2 Table of constants

Symbol	Signification	value	unit
LCR	LiDAR coverage radius	200	m
V_{usv}	USV speed	2	m/s
V_{max}	maximal tidal current speed	1	m/s
ω	tidal current pulsation	$2.\Pi/3600$	1/s
$\beta.V_{usv}^3$	hydrodynamic darg	8	kg.m/s

7.1 Exact vs heuristic method

Table 3 shows the obtained consumed energy and the computation time while solving the covering problem of G_1, G_2, G_3 and G_4 graphs described in Table 1 and illustrated in Fig-7. For both S-PAES, and C-PAES, the number of executed iterations is 10000.

The results shows that for graphs with way-point number lower than 74 (G_3), both heuristics and exact solvers are able to find the same solutions. However, the computation time of the exact solver is much higher than the heuristic one. For example,

Table 3 Exact and heuristic solution for G_1, G_2, G_3 and G_4 graphs and their computation time

Graph	α	Energy (joule)			Computation time (s)		
		Exact	S-PAES	C-PAES	Exact	S-PAES	C-PAES
$G1$	0.2	8000	8000	8000	1357.68	0.38	0.37
	0.4	9600	9600	9600			
	0.6	14400	14400	14400			
	0.8	17600	17600	17600			
	1.0	24800	24800	24800			
G_2	0.2	5760	5760	5760	2242.27	0.52	0.38
	0.4	1080	1080	1080			
	0.6	15840	15840	15840			
	0.8	17280	17280	17280			
	1.0	21600	21600	21600			
G_3	0.2	6000	6000	6000	51333.71	0.72	0.67
	0.4	9600	9600	9600			
	0.6	//	1500	1500			
	0.8	16800	16800	16800			
	1.0	20600	20600	20600			
G_4	0.2	//	6000	6000	∞	1.16	0.96
	0.4	//	12000	12000			
	0.6	//	15000	15000			
	0.8	//	19000	19000			
	1.0	//	24000	24000			

for G_3 graph the exact solver needs more than 14 hours while S-PAES and C-PAES require 0.72 and 0.67 seconds respectively. We can observe that for low complexity problems, the number of iteration executed is sufficient to explore all possible covering tours which explain that the solution found by PAES is the same of the optimal one".

Starting from G_4 graph, we were not able to get any solution with the limit of 24 hours. This case is denoted by ∞ symbol. On the other hand, the energy required to fully ($\alpha = 1$) cover G_1, G_2 and G_3 are 24800, 21600 and 20600 joules respectively. These slight variations are due the changes in graph geometry caused by the variation in vertex distances. We note $E_{avg} = 22366$ the average consumed energy provided by the exact method. We use this reference value hereafter to evaluate the quality of the solutions provided by S-PAES and C-PAES for graph size beyond 110 vertices.

7.2 PAES evaluation

7.2.1 S-PAES vs C-PAES

In Table 4, we consider the coverage of graphs with a higher number of way-points starting from 172 to 4564. Since the exact solver is not able to provide solution within acceptable delay, we only analyse the solution provided by PAES. Notice is that listed solutions correspond to $\alpha = 1$ which means that the coverage objective is 100% met.

In Table 4, we can observe that within the same number of iteration, the execution of C-PAES is faster than that of S-PAES while providing a better solution in term of energy consumption. In fact, as explained in section **??**, the reduction of chromosome size reduces the size of the considered vertices in each iteration leading to lower

Table 4 S-PAES vs C-PAES ($\alpha = 1$, iteration number = 100000)

Graph		Energy (joule)		Computation time (s)	
d (m)	n	S-PAES	C-PAES	S-PAES	C-PAES
100	172	23600	20300	2.11	1.87
90	205	24120	20520	2.88	2.5
80	250	24480	22080	3.95	3.58
70	323	25200	22520	5.71	5.36
60	466	26160	23000	9.83	8.84
50	641	26600	23520	13.87	13.37
40	989	28320	25240	30.36	25.96
30	2029	33500	28200	109.17	75.80
25	2596	34780	29600	158.6	95.85
20	4564	36160	30400	271.17	120.73

computation time. Moreover, it allows reducing the probability of covering a vertex more than once, and consequently, reduces the consumed energy.

On the other hand, we can notice that when the number of vertices is higher (more complex graph), the founded solution tends to diverge from that provided by the exact method (E_{avg}). For example, in case of vertices number n = 2595, the solution provided by C-PAES is 30% greater than E_{avg}. One can suppose that increasing the number of iteration may help converging to desired solution. In what follows, we will analyse the impact of iteration number on the solution quality.

7.2.2 C-PAES convergence

Figure- 8 shows the C-PAES results for different iteration numbers. We notice that increasing the number of iterations reduces the gap between the heuristic solution and the reference value (E_{avg}). For example, the solution provided after 10^6 and 10^7 iterations in case of vertices size when n=2595 is 25% and 7% respectively. Actually, depending on the constraints of the considered problem, the iteration number can be used as a parameter to a achieve the trade-off between the computation time and the solution quality.

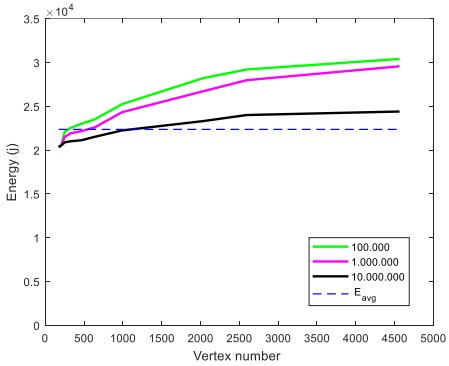

Fig. 8 C-PAES results from different iteration numbers (100% of covered area)

7.3 Study case: Brest city harbor monitoring

Our approach is experimented based on realistic scenario by applying the C-PAES method to monitor Brest port.Figure 9 shows Brest's port map represented with a binary map where land and sea are represented with black and white colors respectively. The approximate harbor surface is $3km^2$ $(3*1km)$. The free navigation surface is graphed with a grid of 1886 vertices marked by red points. The considered distance between two points is fixed by $24.3m$ for horizontal and $22.2m$ for vertical axes. The blue point is chosen as the starting location of the USV. In this case, the USV speed V_{usv} is fixed to $2m/s$ and LiDAR radius is fixed to $200m$.

Figure- 10 shows the archive of initial and final C-PAES results represented as

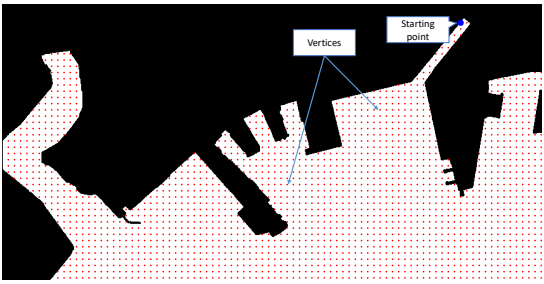

Fig. 9 Binary map representation of the Brest harbor

the Pareto front.The non-dominated Pareto set of solutions acquired by the C-PAES algorithm is shown in red points which represents 100 non-dominated solutions (the archive size fixed for PAES). The green points show the initial PAES archive with 17 non-dominated solutions determined randomly (among 100 generated ones).

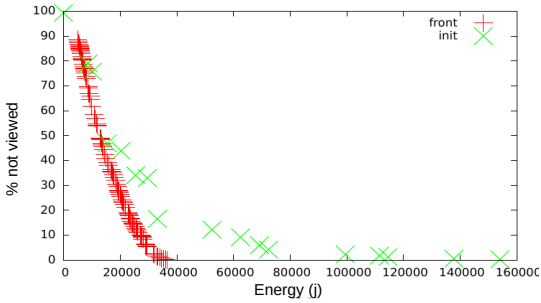

Fig. 10 Initial and final PAES solutions archives

7.3.1 Covering vs consumed energy

Based on the obtained results given in Fig- 10, we extract four solutions that allow USV to cover $25\%, 50\%, 75\%$ and 100% of the harbor respectively. The obtained

paths are illustrated through Fig- 11.a) to Fig- 11.d). The covered area rates and consumed energies are listed in Table 5.

Table 5 Consumed energies for different covered space rates

Scenario	Covered area (%)	Energy (j)
(a)	25%	7485
(b)	50%	12889
(c)	75%	20572
(d)	100%	36942

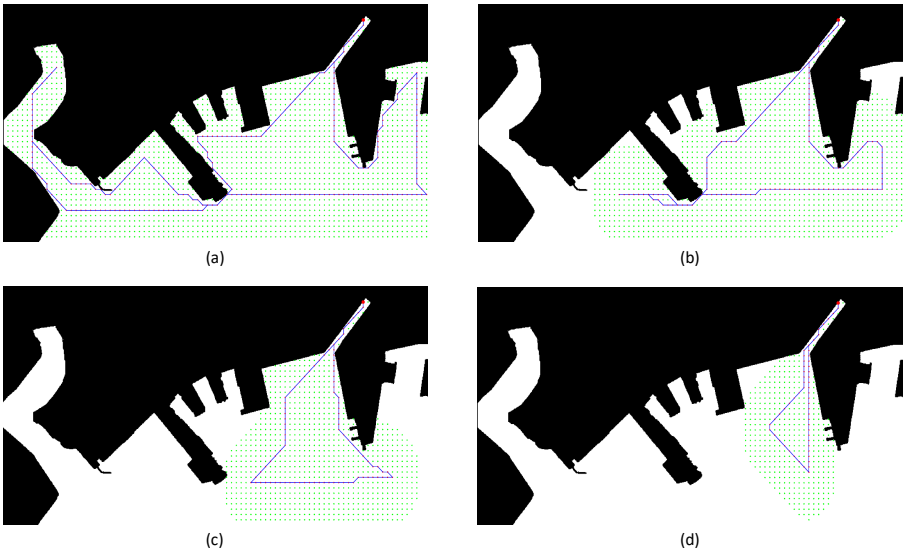

Fig. 11 USV tours with different coverage rates (a.100%, b.75%, c.50%, d.25%)

7.3.2 Impact of USV departure time on the consumed energy for covering the whole harbor

Variations in tidal current velocities over time result in fluctuations on the USV energy consumption during navigation. Table 6 lists 12 USV missions that allow covering the full port with different departure times. It illustrates the tidal currents effects modelled in the given section 5.1.2. $t - start$ is the hour number after the high tide, $Energy$ is the required energy to make the tour that covers the whole harbor, and V_c is the tidal current velocity at starting time. According to Table 6, this approach makes it feasible to choose the best moment to start the USV monitoring. Therefore, in this simulation, the tenth hour after the high tide must be the best instant to start USV monitoring mission, which enables to reduce the energy consumption

by a factor of 42% compared to the worst case. Figure 12 shows the USV followed path allowing to cover the whole space starting at the 10^{th} hour after high tide.

Table 6 Consumed energies for full port coverage versus starting times

t-start(hrs)	**Vc** (m/s)	**Energy** (j)
0	$0,00$	36447
1	$0,49$	48367
2	$0,36$	41758
3	$1,00$	41449
4	$0,86$	48402
5	$0,50$	41084
6	$0,00$	46937
7	$-0,49$	59232
8	$-0,86$	39513
9	$-0,99$	51243
10(*)	$-0,86$	34031
11	$-0,50$	52580

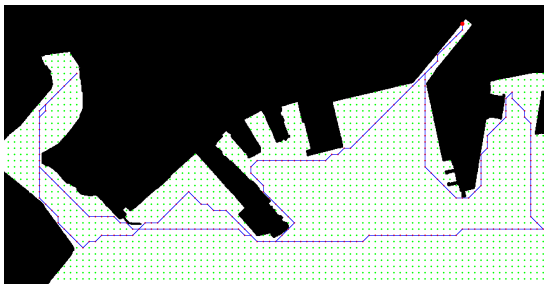

Fig. 12 USV tour starting at 10^{th} hour after the high tide to cover the entire harbor

8 Conclusion

In this paper, we addressed the marine monitoring problem with two conflicting objectives : maximization of the covered area and minimization of the consumed energy. We have shown that this kind of problem cannot be exactly solved within an acceptable delay. This is particularly true in the case of marine monitoring where the environment conditions such as tidal currents may change rapidly. Thus, we have proposed a heuristic approach which provide a good trade-off between solution quality and computation time.

In the heuristic solver, we have considered realistic conditions such as tidal currents which is a very important parameter to be considered when planning a monitoring mission. For example, we have demonstrated that a considerable energy can be saved if we select the USV mission departure time according to the tidal currents effects.

However, one may expect to achieve additional energy saving if we consider the

USV speed as an additional parameter of the objective function. Actually, since the energy model is a cubic function of the USV speed (See Eq.5), the USV may decrease its speed in case of counter-current and increase it in case of co-current [36]. In this case, the tour covering time (depending on the USV speed) will be a new objective which conflicts with the consumed energy and the covered area.
Finally, the proposed solver can be extended to support multi-USV harbour monitoring. In this case, the challenge is to partition optimally the monitored area so it can be covered in parallel with multiple USVs.

Declarations

Funding: No funding was provided for this research.
Conflicts of interest/Competing interests : The authors have no conflicts of interest to declare that are relevant to the content of this article
Ethics approval: No ethical approval was deemed necessary.
Availability of data and material: Data and materials developed in this research are not currently online.
Authors' contributions: All authors H.Ouelmokhtar, Y.Benmoussa, D-P.Diguet, D.Benazzouz and L.Lemarchand, have contributed in softwar development, writing, corrections and improvements of the paper.
Consent to participate: The authors, H.Ouelmokhtar, Y.Benmoussa, D-P.Diguet, D.Benazzouz and L.Lemarchand voluntarily agree to participate in this research study.
Consent for publication: The authors, H.Ouelmokhtar, Y.Benmoussa, D-P.Diguet, D.Benazzouz and L.Lemarchand give their consent for information about themselves to be published in the Journal of Intelligent Robotic Systems.

References

[1] Liu, Z., Zhang, Y., Yu, X., Yuan, C.: Unmanned surface vehicles: An overview of developments and challenges. Annual Reviews in Control **41**, 71–93 (2016)

[2] Zhou, C., Gu, S., Wen, Y., Du, Z., Xiao, C., Huang, L., Zhu, M.: The review unmanned surface vehicle path planning: Based on multi-modality constraint. Ocean Engineering **200**, 107043 (2020)

[3] Steimle, E.T., Hall, M.L.: Unmanned surface vehicles as environmental monitoring and assessment tools. In: OCEANS 2006, pp. 1–5 (2006). IEEE

[4] Pastore, T., Djapic, V.: Improving autonomy and control of autonomous surface vehicles in port protection and mine countermeasure scenarios. Journal of Field Robotics **27**(6), 903–914 (2010)

[5] Shafer, A.J., Benjamin, M.R., Leonard, J.J., Curcio, J.: Autonomous cooperation of heterogeneous platforms for sea-based search tasks. In: OCEANS 2008,

pp. 1–10 (2008). https://doi.org/10.1109/OCEANS.2008.5152100

[6] Lazarowska, A.: A new deterministic approach in a decision support system for ship's trajectory planning. Expert Systems with Applications **71**, 469–478 (2017)

[7] Lenes, J.H.: Autonomous online path planning and path-following control for complete coverage maneuvering of a usv. Master's thesis, NTNU (2019)

[8] Golden, B., Naji-Azimi, Z., Raghavan, S., Salari, M., Toth, P.: The generalized covering salesman problem. INFORMS Journal on Computing **24**(4), 534–553 (2012)

[9] Salari, M., Naji-Azimi, Z.: An integer programming-based local search for the covering salesman problem. Computers & Operations Research **39**(11), 2594–2602 (2012)

[10] Salari, M., Reihaneh, M., Sabbagh, M.S.: Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem. Computers Industrial Engineering **83**, 244–251 (2015). https://doi.org/10.1016/j.cie.2015.02.019

[11] Pandiri, V., Singh, A., Rossi, A.: Two hybrid metaheuristic approaches for the covering salesman problem. Neural Computing and Applications **32**(19), 15643–15663 (2020)

[12] Tripathy, S.P., Biswas, A., Pal, T.: A multi-objective covering salesman problem with 2-coverage. Applied Soft Computing **113**, 108024 (2021)

[13] Reinhelt, G.: {TSPLIB}: a library of sample instances for the tsp (and related problems) from various sources and of various types. URL: http://comopt. ifi. uniheidelberg. de/software/TSPLIB95 (2014)

[14] Wolsey, L.A.: Mixed integer programming. Wiley Encyclopedia of Computer Science and Engineering, 1–10 (2007)

[15] Croes, G.A.: A method for solving traveling-salesman problems. Operations research **6**(6), 791–812 (1958)

[16] Ouelmokhtar, H., Benmoussa, Y., Benazzouz, D., Ait-Chikh, M.A., Lemarchand, L.: Energy-based usv maritime monitoring using multi-objective evolutionary algorithms. Ocean Engineering **253**, 111182 (2022)

[17] Current, J.R., Schilling, D.A.: The covering salesman problem. Transportation science **23**(3), 208–213 (1989)

[18] Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Operations research **21**(2), 498–516 (1973)

[19] Zhang, W., Xu, Y., Xie, J.: Path planning of usv based on improved hybrid genetic algorithm. In: 2019 European Navigation Conference (ENC), pp. 1–7 (2019). IEEE

[20] Kim, H., Kim, S.-H., Jeon, M., Kim, J., Song, S., Paik, K.-J.: A study on path optimization method of an unmanned surface vehicle under environmental loads using genetic algorithm. Ocean Engineering **142**, 616–624 (2017)

[21] Ma, Y., Hu, M., Yan, X.: Multi-objective path planning for unmanned surface vehicle with currents effects. ISA transactions **75**, 137–156 (2018)

[22] Song, C.H.: Global path planning method for usv system based on improved ant colony algorithm. In: Applied Mechanics and Materials, vol. 568, pp. 785–788 (2014). Trans Tech Publ

[23] Xia, G., Han, Z., Zhao, B., Liu, C., Wang, X.: Global path planning for unmanned surface vehicle based on improved quantum ant colony algorithm. Mathematical Problems in Engineering **2019** (2019)

[24] Niu, H., Lu, Y., Savvaris, A., Tsourdos, A.: An energy-efficient path planning algorithm for unmanned surface vehicles. Ocean Engineering **161**, 308–321 (2018)

[25] Singh, Y., Sharma, S., Sutton, R., Hatton, D., Khan, A.: A constrained a* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. Ocean Engineering **169**, 187–201 (2018)

[26] Singh, Y., Sharma, S., Sutton, R., Hatton, D.: Towards use of dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field. TransNav, International Journal on Marine Navigation and Safety od Sea Transportation **12**(1) (2018)

[27] Zhang, J., Zhang, F., Liu, Z., Li, Y.: Efficient path planning method of usv for intelligent target search. Journal of Geovisualization and Spatial Analysis **3**(2), 13 (2019)

[28] Song, R., Liu, Y., Bucknall, R.: Smoothed a* algorithm for practical unmanned surface vehicle path planning. Applied Ocean Research **83**, 9–20 (2019)

[29] Touzout, W., Benmoussa, Y., Benazzouz, D., Moreac, E., Diguet, J.-P.: Unmanned surface vehicle energy consumption modelling under various realistic disturbances integrated into simulation environment. Ocean engineering **222**, 108560 (2021)

[30] Park, J., Kim, S., Noh, G., Kim, H., Lee, D., Lee, I.: Mission planning and performance verification of an unmanned surface vehicle using a genetic algorithm.

International Journal of Naval Architecture and Ocean Engineering (2021)

[31] Zhang, W., Wang, K., Wang, S., Laporte, G.: Clustered coverage orienteering problem of unmanned surface vehicles for water sampling. Naval Research Logistics (NRL) **67**(5), 353–367 (2020)

[32] Knowles, J.D., Corne, D.W.: Approximating the nondominated front using the pareto archived evolution strategy. Evolutionary computation **8**(2), 149–172 (2000)

[33] Dantzig, G.e.a.: Solution of a large-scale traveling-salesman problem. Journal of the operations research society of America **2**(4) (1954)

[34] Wesselmann, F.: Generating general-purpose cutting planes for mixed-integer programs. PhD thesis, PhD thesis, Universität Paderborn, 2011. Cited on (2010)

[35] IBM: ILOG CPLEX Optimization toolkit. http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud. 12/29/2020 (2020)

[36] Caharija, W., Pettersen, K.Y., Gravdahl, J.T.: Counter-current and co-current guidance of underactuated unmanned marine vehicles. IFAC Proceedings Volumes **46**(10), 60–66 (2013)