Area Coverage in Heterogeneous Multistatic Sonar Networks: A Simulated Annealing Approach

Owein Thuillier, Nicolas Le Josse, Alexandru-Liviu Olteanu, Marc Sevaux, Hervé Tanguy

Area Coverage in Heterogeneous Multistatic Sonar Networks: a Simulated Annealing Approach

Owein Thuillier[1,2][0000−0002−1880−9639], Nicolas Le Josse[1], Alexandru-Liviu Olteanu[2][0000−0002−0715−2168], Marc Sevaux[2][0000−0003−3855−9072], and Hervé Tanguy[1][0000−0002−4904−0535]

[1] Thales, Defense and Mission Systems, Brest, France
{owein.thuillier, nicolas.lejosse, herve.tanguy}@fr.thalesgroup.com
[2] Université Bretagne-Sud, Lab-STICC, UMR CNRS 6285, Lorient, France
{owein.thuillier, alexandru.olteanu, marc.sevaux}@univ-ubs.fr

Abstract. In this paper, we propose a Simulated Annealing (SA) metaheuristic for efficiently solving the Area Coverage (AC) problem in Heterogeneous Multistatic Sonar Networks (HMSNs). In this problem, which is new in the literature, the aim is to determine the optimal location for the various sensors making up an HMSN, which is a particular case of Heterogeneous Wireless Sensor Networks (HWSNs). HMSNs are made up of a set of acoustic buoys, or sonobuoys, dropped by an airborne carrier and which can be transmitter-only (Tx), receiver-only (Rx) or transmitter-receiver (TxRx). In particular, an HMSN consists of a set of active sonar systems in monostatic and/or bistatic configuration. The bistatic case refers to the pairing between a source from a Tx or TxRx buoy with a receiver from another Rx or TxRx buoy, which may be kilometers apart from the former. This contrasts with the monostatic case, referring to the situation where both the source and receiver are integrated within the same unit, specifically a TxRx buoy. In addition, in this work, we take into account a certain number of operational aspects such as coastlines, probabilistic detection models, an adverse effect called direct blast as well as variable performance depending on the source/receiver pair under consideration. Morever, we also consider the possibility that some pairings might be infeasible due to inter-sensor incompatibility, for example because of different operating frequencies. Finally, we present numerical results in which we compare ourselves with a set of tailored Mixed-Integer Linear Programs (MILPs) for the same problem.

Keywords: OR in defense · Multistatic sonar networks · Heterogeneous sensors · Area coverage problem · Metaheuristic · Simulated annealing.

1 Introduction

Sonar (SOund NAvigation and Ranging) is an essential asset in the anti-submarine arsenal, as it provides a cutting-edge capability for probing the oceans. In particular, they can be used for underwater target search and tracking operations,

as well as for surveillance or sanctuarization of strategic regions. These systems can be passive, simply listening to the sounds emitted by objects in the vicinity; or active, operating by emitting acoustic waves that reflect off any object within their range, thereby permitting their detection and location [14]. In this study, we focus primarily on active sonar systems, with a particular emphasis on acoustic buoys, commonly known as sonobuoys [7]. These buoys are deployed within the designated Area of Interest (AoI) from an airborne carrier, which could be a Maritime Patrol Aircraft (MPA), a helicopter, or even an Unmanned Aerial Vehicle (UAV). Upon impact with the water surface, their main unit submerges to a predetermined depth, while remaining tethered to a surface float through a wire cable. This float, fitted with a radio antenna, establishes and maintains the communication link with the carrier. Acoustic buoys can be classified into three primary categories: those that are solely transmitters (Tx), exclusively receivers (Rx), and those combining both functions as transmitter-receivers (TxRx). A sonar system is then virtually formed by the pairing of a source from a Tx or TxRx buoy and a receiver from a TxRx or Rx buoy. When both the source and the receiver are co-located, i.e. located within the same physical buoy (i.e. TxRx), we refer to monostatism[3]; whereas we refer to bistatism when they are not co-located, sometimes located several kilometers apart and potentially on two separate immersion planes. Figure 1 illustrates this in a highly simplified way. Furthermore, within each of these three categories of acoustic buoys (Tx, Rx, TxRx), there are several types of buoy whose operating frequencies and performance when paired with other buoys can vary considerably, hence the interest in the heterogeneous problem addressed here [7]. A collection of heterogeneous sonar systems in bistatic and/or monostatic configuration then constitutes a Heterogeneous Multistatic Sonar Network (HMSN).

In the problem that interests us here, the objective is to determine the optimal placement of a set of heterogeneous acoustic buoys over an AoI (forming an HMSN) in order to maximize the coverage rate. This is an Area Coverage (AC) problem in Wireless Sensor Networks (WSNs) [8] and a variant of the Maximal Covering Location Problem (MCLP) [1] if we consider that customers are the targets to be monitored and warehouses are the sonars to be deployed. Even if we simplify the problem by considering only homogeneous TxRx buoys, it is still an $\mathcal{NP}$-Hard problem [10]. Preliminary work addresses the problem in the homogeneous case [3,5,13] and this paper is a direct follow-up to the work of [12] presenting a set of 9 Mixed-Integer Linear Programs (MILPs) for this problem. As in the latter paper, we take into account coastlines, probabilistic detection models, the adverse effect known as direct blast [2], variable performance depending on the source/receiver pair as well as possible incompatible sonar systems (e.g. in the case of different operating frequencies).

[3] The term quasi-monostatism is also used whenever the source and receiver are very close to each other (geographically speaking), although located in two separate buoys.

In this work, we present a metaheuristic, i.e. an approximate resolution method, known as Simulated Annealing (SA) [9] in order to deal more efficiently with this problem. This metaheuristic is based on a set of dedicated neighborhoods, which we describe in more detail below. Moreover, we also propose original visual representations of the solutions which did not exist in the current literature in order to help decision-makers.

The paper is organized as follows. In the first section, we formally present the problem under study. In the second section, we present the proposed resolution method with all the parameters and neighborhoods that we use. In the third section, we present numerical results comparing ourselves with a set of efficient MILPs from the literature. Finally, in the last section, we conclude this work and give some prospects for future research.

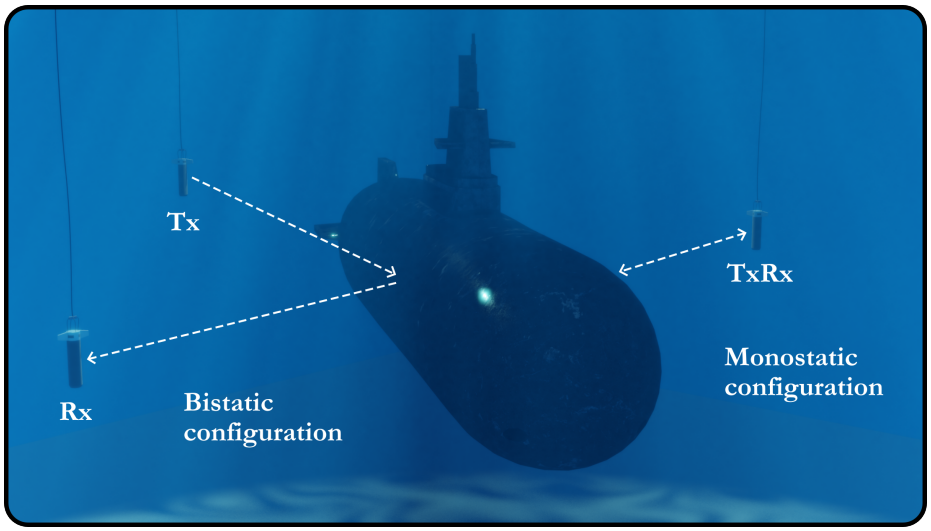


Fig. 1. Simplified illustration of the operational context with a sonar system in bistatic configuration comprised of two sonobuoys on the left side of the submarine (Tx $\rightarrow$ Target $\rightarrow$ Rx) and a sonar system in monostatic configuration on the right side (TxRx $\leftrightarrow$ Target).

2 Problem Description

Given an AoI, we first retrieve a Digital Elevation Model (DEM) of this zone, i.e. a discretization into rectangular regular cells of size $cellsize \in \mathbb{R}^+$ from bathymetric and altimetric data. We thus have $m \in \mathbb{N}^*$ maritime cells, in the center of which we assign a deployment position and a fictitious target used to evaluate the performance of the network at a given point. Formally, we then have $E = \{e_1, \ldots, e_m\} \subseteq \mathbb{R}^2$ the set of maritime cells and $T = \{t_1, \ldots, t_m\} \subseteq \mathbb{R}^2$ the set of targets. We denote $TxRx$ the set of buoy types in the TxRx category, Tx

the set of buoy types in the Tx category and Rx the set of buoy types in the Rx category. In addition, we introduce $I = TxRx \cup Tx$ the set of buoy types equipped with a source, $J = TxRx \cup Rx$ the set of buoy types equipped with a receiver and $K = TxRx \cup Tx \cup Rx$ the set of all buoy types. Furthermore, we introduce $n_i \in \mathbb{N}^*$, the quantity of buoys of type $i \in K$ available for deployment and $C \subseteq I \times J$ the set of functional sonar systems, in the sense of compatibility between source and receiver (i.e. operating frequency). The set of admissible solutions (network) to the problem under study, denoted Ω, is then formally defined as

$$\begin{aligned}\Omega = \{\omega \subseteq E \times K \mid \forall e \in E, |\{(e,i) \in \omega\}| \leq 1, \textbf{(a)} \\ \forall i \in K, |\{(e,i) \in \omega\}| \leq n_i, \textbf{(b)}\},\end{aligned} \tag{1}$$

where **(a)** forces a maximum of one buoy per position and **(b)** is a constraint on the number of buoys deployed for each type. Next, we introduce the set of all theoretically possible sonar systems, denoted Ξ, formally defined as

$$\Xi = \{(s,r) \mid s = (e,i) \in E \times I, r = (e',j) \in E \times J, (i,j) \in C\}, \tag{2}$$

By extension, we will thus denote $\Xi_\omega \subseteq \Xi$ the set of sonar systems in a network $\omega \in \Omega$. Concerning the detection model, we now introduce $P_d^\omega(t)$, the Cumulative Detection Probability (CDP) of the target t by a network $\omega \in \Omega$, computed as follows:

$$P_d^\omega(t) = 1 - \prod_{(s,r) \in \Xi_\omega} \left(1 - P_d^{(s,r)}(t)\right), \tag{3}$$

where $P_d^{(s,r)}(t)$ is the Instantaneous Detection Probability (IDP) of the target t. It is calculated using the following Fermi function (sigmoid-type):

$$P_d^{(s,r)}(t) = \frac{1}{1 + 10^{\frac{\left(\frac{\rho_{t,s,r}}{\rho_0}\right) - 1}{b}}}, \tag{4}$$

with $\rho_{t,s,r} = \sqrt{d_{s,t} d_{t,r}}$ and where $d_{s,t}$ and $d_{t,r}$ are respectively the source-to-target and target-to-receiver distances. For coastline management, this IDP will be set to 0 if the discretization of one of the two source $\rightarrow$ target or receiver $\rightarrow$ target segments crosses a terrestrial cell. We also take into account an undesirable effect called direct blast, which makes detection theoretically impossible inside an ellipse of equation:

$$d_{s,t} + d_{t,r} < d_{s,r} + 2r_b, \tag{5}$$

with $r_b = \frac{c\tau}{2}$ (in km) and which is equal to half the "pulse length", i.e. the distance travelled by the acoustic wave during the period $\tau \in \mathbb{R}^+$ (in s) at celerity $c \in \mathbb{R}^+$ (in $km \cdot s^{-1}$). A target will then be considered as covered (detected) when the CDP is greater than a threshold $\phi \in [0,1]$ fixed beforehand and generally

close to 1, i.e. whenever $P_d^{\omega}(t) \geq \phi$. For more details on these detection models, the reader is invited to refer to [13,12]. Finally, the objective function of the problem corresponds here to the coverage rate of the AoI, formally defined as:

$$\begin{aligned} f\colon \Omega &\to [0,1] \\ \omega &\mapsto \frac{1}{|T|} \sum_{t \in T} \sigma\left(P_d^{\omega}(t)\right), \end{aligned} \tag{6}$$

where

$$\sigma(x) = \begin{cases} 1 \text{ if } x \geq \phi, \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

Ultimately, we attempt to find the optimal solution $\omega^* \in \Omega$, i.e. the HMSN maximising the function f previously defined:

$$\omega^* = \arg\max_{\omega \in \Omega} f(\omega). \tag{8}$$

3 Simulated Annealing (SA)

In this section, we present the SA [9] that we have implemented based on the component-based classification proposed in [4]. For the complete pseudo-code of the SA, readers are invited to consult the latter paper. We thus begin with the two components that are problem-specific and end with the 7 components that are algorithm-specific. As far as implementation is concerned, we rely heavily on the techniques presented in [13] to be computationally frugal, and adapted here to the heterogeneous case.

3.1 Problem-specific components

Initial Solution. The initial solution $\omega_0 \in \Omega$ provided to the SA is constructed here by randomly positioning all available sonobuoys on the AoI. Although it would be possible to build this initial solution iteratively, for example using a greedy heuristic such as the one presented in [13], the idea of an initial random solution (like in [6]) is to introduce some variability at the initialization of the algorithm. This variability at initialization is particularly useful when several runs are performed (multi-start).

Neighborhoods. In this section, we introduce two different neighborhoods families based on a given $\omega \in \Omega$ solution: *k-n-m-shift* and *k-swap*. The way in which these neighborhoods will be used will be detailed in Section 3.2 with the exploration component. Moreover, for these neighborhoods, updating the objective function does not require a complete recalculation of the CDPs, but only an adjustment for the buoys that have been moved. The first family, called *k-n-m-shift*, corresponds to shifting a set of k sensors in a discrete square annulus

region delimited by the inner ring n and the outer ring m. For a better understanding, the reader is referred to Figure 2 for an illustration of the ring concept and to Figure 3 for examples of movements with this specific neighborhood. Formally, starting from a solution $\omega \in \Omega$, with $1 \leq k \leq |\omega|$, $n \geq 1$, $m \geq n$, the neighborhood of ω with this family is defined as follows:

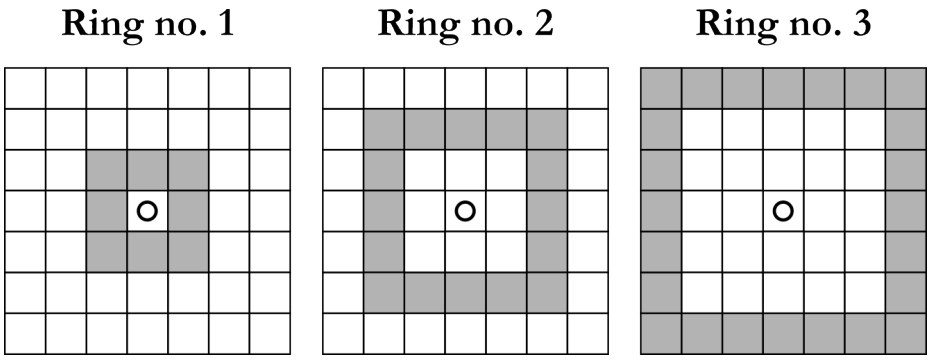


Fig. 2. Illustration of the ring concept surrounding a given deployment position.

$$\mathcal{N}_{shift}^{k-n-m}(\omega) = \Big\{\omega' \in \Omega \mid |\omega| = |\omega'|, \ \textbf{(a)}$$
$$\exists B_1 \subseteq \omega, \exists B_2 \subseteq \omega, |B_2| = k \wedge B_1 \cap B_2 = \emptyset \wedge B_1 \cup B_2 = \omega, \ \textbf{(b)}$$
$$\forall b \in B_1, \exists b' \in \omega', b = b', \ \textbf{(c)}$$
$$\forall b = (e,i) \in B_2, \exists b' = (e',i') \in \omega', i = i' \wedge e' \in \mathcal{A}^{n-m}(e) \ \textbf{(d)}\Big\},$$

where: **(a)** forces the original solution ω and the neighboring solution ω' to have the same number of sensors (buoys); **(b)** is a partition of the original solution into two subsets of buoys: B_1 the set of $|\omega| - k$ buoys that will not be shifted and B_2 the set of k buoys that will be shifted; **(c)** all the buoys belonging to the subset B_1 must be present in the neighboring solution ω', as they are not shifted; **(d)** all the buoys $b = (e,i)$ in the subset B_2 must be found in the neighboring solution ω' at one of the positions in the discrete square annulus region around their original positions e. The set of these eligible positions is denoted by the set $\mathcal{A}^{n-m}(e)$, which is formally defined as:

$$\mathcal{A}^{n-m}(e = (x,y)) = \Big\{e' = (x',y') \mid \exists \Delta_x \in [\![-m,m]\!], \exists \Delta_y \in [\![-m,m]\!],$$
$$|\Delta_x| \in [\![n,m]\!] \vee |\Delta_y| \in [\![n,m]\!],$$
$$x' = x + \Delta_x \cdot cellsize \wedge y' = y + \Delta_y \cdot cellsize\Big\}.$$

Note that it is virtually impossible to shift a buoy to a position already occupied by another buoy with this type of movement (implicit by the definition of Ω). An upper bound on the size of this neighborhood is:

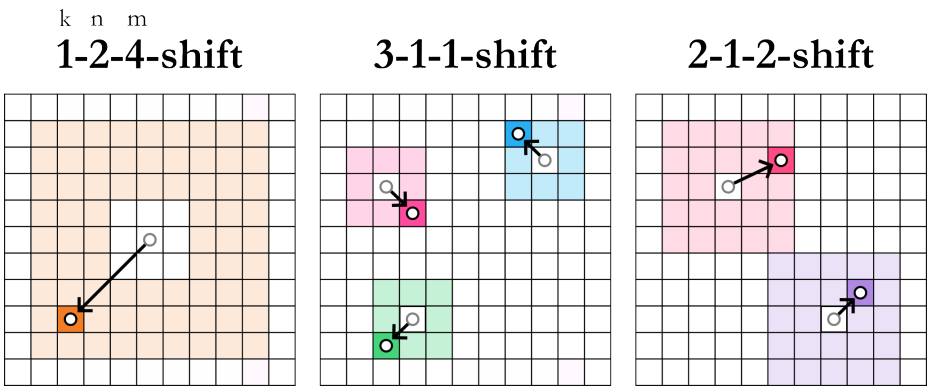


Fig. 3. Illustration of the *k-n-m-shift* movement family, with k the number of sensors to be shifted, n the inner ring and m the outer ring, defining a discrete square annulus region as potential landing positions (color-coded cells).

$$|\mathcal{N}_{shift}^{k-n-m}(\omega)| \leq \frac{|\omega|!}{(|\omega|-k)!k!} \cdot \left(\sum_{i \in [\![n,m]\!]} 8i \right)^k, \tag{9}$$

which can be tightened slightly by excluding positions that are already occupied and taking into account terrestrial cells. The second family, called *k-swap*, involves exchanging the two sensors within k unique pairs (i.e. no overlap between pairs). One particularity is that it is not permitted to have two sensors of the same type in a given pair, as this is of no interest in terms of coverage payoff. Examples of movements with this specific neighborhood are available in Figure 4. Formally, starting from a solution $\omega \in \Omega$, with $1 \leq k \leq \left\lfloor \frac{|\omega|}{2} \right\rfloor$, the neighborhood of ω with this family is defined as follows:

$$\begin{aligned}
\mathcal{N}_{swap}^{k}(\omega) = \Big\{ & \omega' \in \Omega \mid |\omega| = |\omega'|, \ \textbf{(a)} \\
& \exists B \subseteq \omega, \forall i \in [\![1,k]\!], \exists B_i = (b_1 = (e_1, i_1), b_2 = (e_2, i_2)) \in \omega, i_1 \neq i_2, \\
& B \cup \bigcup_{i \in [\![1,k]\!]} B_i = \omega \wedge \\
& \forall i \in [\![1,k]\!], B \cap B_i = \emptyset \wedge \forall (i,j) \in [\![1,k]\!]^2, i \neq j, B_i \cap B_j = \emptyset, \ \textbf{(b)} \\
& \forall b \in B, \exists b' \in \omega', b = b', \ \textbf{(c)} \\
& \forall i \in [\![1,k]\!], \forall (b_1 = (e_1, i_1), b_2 = (e_2, i_2)) \in B_i, \exists (b_1' = (e_1', i_1'), b_2 = (e_2', i_2')) \in \omega' \\
& e_1' = e_2 \wedge i_1' = i_1 \wedge e_2' = e_1 \wedge i_2' = i_2 \ \textbf{(d)} \Big\}
\end{aligned}$$

where: **(a)**, as before, forces the original solution ω and the neighboring solution ω' to have the same number of sensors; **(b)** is a partition of the original solution into a first subset B of buoys which will not be moved and k subsets

of two buoys of different types whose positions will be interchanged (swapped); **(c)** all the buoys belonging to set B must be present as such in the neighboring solution; **(d)** for each subset of two buoys in the partition of the original ω solution, the same two buoys must be found in the neighboring solution with their positions interchanged. An upper bound on the size of this neighborhood is:

$$|\mathcal{N}^k_{swap}(\omega)| \leq \prod_{i \in [\![1,k]\!]} \frac{(|\omega| - 2(i-1))!}{(|\omega| - 2i)!2}, \tag{10}$$

which can be tightened by removing pairs of buoys of the same type, as they have no interest in being exchanged, as discussed earlier.

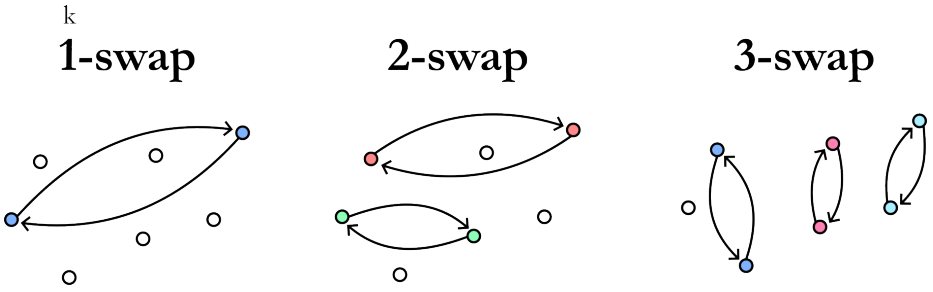


Fig. 4. Illustration of the *k-swap* movement family, where k is the number of unique pairs within which the two sensors will be swapped.

3.2 Algorithm-specific components

Here, we present the 7 algorithm-specific components derived from the classification presented in [4] and summarized in Table 3.2. First, we introduce $\Delta(\omega, \omega') = f(\omega') - f(\omega)$: the difference in coverage rate between two candidate solutions $\omega, \omega' \in \Omega$.

Acceptance Criterion (AC). We have chosen the Metropolis-based criterion proposed in the original formulation of the SA [9] and denoted **AC1**. This criterion systematically accepts movements that lead to improving or same quality solutions, i.e. whenever $\Delta(\omega, \omega') \geq 0$ (maximization problem), and accepts deteriorating movements with a certain probability depending on the ratio between the value $\Delta(\omega, \omega')$ and the current temperature T. Formally:

$$p_{\text{METROPOLIS}} = \begin{cases} 1 & \text{if } \Delta(\omega, \omega') \geq 0, \\ e^{\frac{\Delta(\omega,\omega')}{T}} & \text{otherwise.} \end{cases} \tag{11}$$

Stopping Criterion (SC). As a stopping criterion for the SA algorithm, we have chosen a maximum number of iterations, or candidate moves, denoted by

SC2. Contrary to a fixed computational budget as a stopping criterion, the execution time is not determined *a priori* but rather depends on the search progress, which in turn depends directly on the hardware involved and the instance under consideration. Here, the maximum number of iterations is fixed at 30 000.

Initial Temperature (IT). The initial temperature T_0 is derived from an initial acceptance probability of deteriorating movements, denoted p_0, which correspond to **IT6**. To do this, we perform an initial random walk in the search space, which gives us a sequence of admissible solutions $\omega_0, \omega_1, \ldots, \omega_l \in \Omega$ with $l \in \mathbb{N}^*$ the total length of the walk. From the values $f(\omega_0), f(\omega_1), \ldots, f(\omega_l)$ which we treat as a time series, we derive Δ_{avg} the average gap between two consecutive solutions in the random walk. Then, using the Metropolis criterion, we find $T_0 = \frac{\Delta_{avg}}{ln(p_0)}$. Here, we set l to 5 % of the total number of iterations, i.e. 1 500, and p_0 to 40 % with the aim of achieving a good balance between intensification (hill-climbing behavior, $p_0 = 0.0$) and diversification (random-walk behavior, $p_0 = 1.0$). In addition, we retain the best solution found during this random walk as the initial solution of the SA algorithm.

Exploration Criterion (EC). For the exploration criterion, we have chosen to randomly explore the neighborhood $\mathcal{N}(\omega)$ of a solution $\omega \in \Omega$, which means that at each iteration we generate and evaluate a solution drawn randomly from this neighborhood. This corresponds to **EC1**[4], but we here propose a variant. Indeed, in our case, we define $\mathcal{N}(\omega) = \mathcal{N}_{shift}^{1-1-1}(\omega) \cup \mathcal{N}_{shift}^{2-1-1}(\omega) \cup \mathcal{N}_{shift}^{3-1-1}(\omega) \cup \mathcal{N}_{swap}^{1}(\omega)$ and the drawing will thus be made through a roulette wheel selection with a uniform distribution between the 4 sub-neighborhoods.

Temperature Length (TL). Let $L \in \mathbb{N}^*$ be the number of iterations performed at a given temperature, i.e. before updating the temperature according to the chosen cooling scheme. Here we have chosen to carry out a single iteration at each temperature, which therefore corresponds to **TL1** with $L = 1$ in the "fixed temperature length" category.

Cooling Scheme (CS). For temperature updating, we have chosen a geometric scheme, like the original paper [9]. In this scheme, the temperature at iteration $i+1$ is calculated as follows: $T_{i+1} = \alpha \cdot T_i$ with $0 < \alpha < 1$ and generally α close to 1 so as to have a slow monotonic decrease in temperature, making the acceptance of deteriorating movements increasingly unlikely. This therefore corresponds to **CS2** and we set $\alpha = 0.999$ in our case.

Temperature Restart (TR). Finally, in order to avoid converging towards hill-climbing behavior as the temperature gradually decreases, we will perform restarts. The hope behind this choice is to escape a local optimum that could have been reached at a certain point during the search. To do this, we calculate the overall average acceptance rate among deteriorating movements and we perform

[4] In the original paper, this is NE1, but we prefer EC1 for reasons of uniformity.

a restart when this rate falls below a given threshold, which corresponds to **TR6**. When restarting, the best solution found so far is used as a starting point and the acceptance rate is reset to 0. Furthermore, we here propose a variant where we must wait for $n \in \mathbb{N}^*$ iterations before restarting, so that the acceptance rate becomes reasonably representative of the current situation. In our case, we set the threshold at 5 % and the minimum number of iterations at 100.

Acceptance Criterion (AC)	Stopping Criterion (SC)	Initial Temperature (IT)	Exploration Criterion (EC)	Temperature Length (TL)	Cooling Scheme (CS)	Temperature Restart (TR)
AC1	**SC2**	**IT6**	**EC1**	**TL1**	**CS2**	**TR6**

Table 1. Characterization of the Simulated Annealing (SA) metaheuristic according to the classification proposed in [4]: 7 algorithm-specific components.

4 Numerical Experiments

4.1 Instances

First of all, concerning the instances, we have selected a subset of 32 instances from the 100 instances presented in [12] and detailed exhaustively in the following GitHub directory: https://github.com/owein-thuillier/MSN-dataset (last access: April 1st, 2024). More precisely, we have selected 8 difficult instances in each of the 4 main groups (i.e. DEMs): peninsula, strait, island and river. A given instance then corresponds to a DEM from which it is derived, along with a volume of sensors for each of the types listed in Table 2 below. The inter-sensor performances $\rho_0^{i,j}$ (km) are presented in the double-entry Table 3 and the parameters used in these experiments are listed in Table 4. Given performances are example values for the purpose of demonstration, yet realistic enough.

4.2 Results

The numerical results are reported in Table 4.2, where, for the 32 instances, we have: **(a)** the coverage rate (%) and the resolution time (s) of the best solution found by the best of the 9 MILPs presented in [12]; **(b)** statistics on the 20 SA executions, i.e. average (avg.), standard deviation (std.), minimum (min.) and maximum (max.) for both coverage rate (%) and execution times (s). Note that the presence of an asterix next to the coverage rate means that this is the optimal solution, proven by the execution of one of the MILPs. Besides, a coverage rate in bold means that it is the best integer solution known to date, which may not be optimal (or proven optimal). All the experimentations have been carried out on a Debian 11 server, 64-bit architecture, equipped with 190 GB of RAM and 2 Intel® Xeon® Gold 6258R processors running at a clock speed of 2.70 GHz, each having 50 cores. Moreover, the implementations were done under Julia 1.7.3 and the exact resolutions were performed using IBM ILOG CPLEX 20.1 with default settings, 8 threads in parallel and a computational budget of 7 200 seconds (2 hours). No parrallelism was set up for the SA.

Category	TxRx		Tx		Rx			
Type	A_{HF}	B_{LF}	C_{HF}	D_{LF}	E_{HF}	F_{HF}	G_{LF}	H_{LF}

Table 2. Types (name and operating frequency) of the different sonobuoys considered here in each of the three categories: TxRx, Tx and Rx. HF stands for High Frequency and LF for Low Frequency.

I \ J	A_{HF}	E_{HF}	F_{HF}	B_{LF}	G_{LF}	H_{LF}
A_{HF}	5.0	4.0	3.5	x	x	x
C_{HF}	4.5	3.0	2.5	x	x	x
B_{LF}	x	x	x	8.0	7.5	6.5
D_{LF}	x	x	x	7.0	6.0	5.5

Table 3. Performances ($\rho_0^{i,j}$, in km) of the various compatible sonar systems (i.e. $(i,j) \in C$).

c	τ	r_b	ϕ	Fermi (b)
1.5	1	0.75	0.95	0.2

Table 4. Parameter values for numerical experiments.

If we take the example of instance no. 6 in the "peninsula" group, we see that the best MILP model terminates at the end of the computational budget of 7 200 seconds and returns an integer solution with a coverage rate of 85.71 %. The SA, over 20 runs, found an average coverage rate of 85.57 % with an average execution time of 1.467 s, i.e. around 4908 times faster for only 0.14 percentage points less in coverage rate. In addition, the minimum coverage rate is 84.29 %, the maximum coverage rate is 85.71 % (best known solution) and the standard deviation is 0.44 %. In terms of resolution times, we have a minimum of 1.439 s, a maximum of 1.511 s and a standard deviation of 0.020 s. Finally, based on all the results, we can see that the SA gives excellent results and is highly robust, both in terms of execution times and coverage rates. Indeed, on average across all instances, the SA is 3253 times faster for only 0.59 percentage points less in coverage rate. Besides, the solutions obtained are also robust, as they have a mean standard deviation of 0.036 s for execution times and 0.60 % for coverage rates. Furthermore, the best known integer solution has been found at least once in the entirety of the instances, and often on several runs (sometimes finding better solutions, such as instance no. 13).

Future work should focus on studying and improving the performance of this metaheuristic on a wider range of instances, in particular by varying the sizes and geometric configurations of the AoIs, using, for example, the catalogue of 17 700 instances provided in [11].

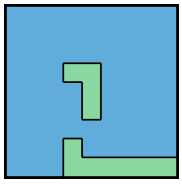
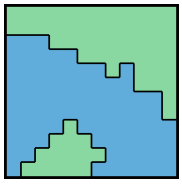
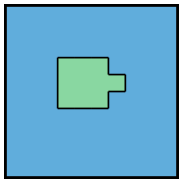
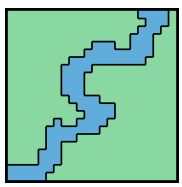
Group	Instance	Best MILP		Simulated Annealing (SA) - 20 runs							
				CPU time (s)				Coverage (%)			
		CPU time (s)	Coverage (%)	avg.	std.	min.	max.	avg.	std.	min.	max.
PENINSULA											
 *9 × 9 (81 cells)* ***70*** *maritime (86.42 %)* *11 terrestrial (13.58 %)*	06	7200.00	**85.71**	1.467	0.020	1.439	1.511	85.57	0.44	84.29	**85.71**
	08	3295.51	**72.86***	1.347	0.046	1.304	1.528	72.86	0.00	72.86	**72.86***
	11	7200.00	**80.00**	1.310	0.010	1.285	1.327	79.86	0.44	78.57	**80.00**
	13	7200.00	92.86	2.432	0.034	2.382	2.523	92.79	0.73	91.43	**94.29**
	17	3266.10	**94.29***	1.707	0.011	1.694	1.744	94.14	0.44	92.86	**94.29***
	18	7049.85	**98.57***	1.893	0.023	1.857	1.936	96.93	0.84	95.71	**98.57***
	19	7200.00	**91.43**	2.415	0.022	2.376	2.454	91.21	0.52	90.00	**91.43**
	20	7200.00	**67.14**	2.285	0.028	2.242	2.362	66.86	0.59	65.71	**67.14**
STRAIT											
 *12 × 12 (144 cells)* ***81*** *maritime (56.25 %)* *63 terrestrial (43.75 %)*	27	7200.00	**92.59**	1.803	0.156	1.664	2.250	92.59	0.00	92.59	**92.59**
	28	7200.00	**97.53**	1.852	0.019	1.826	1.903	97.53	0.00	97.53	**97.53**
	32	7200.00	**83.95**	2.194	0.038	2.161	2.343	83.40	0.63	82.72	**83.95**
	33	7200.00	**88.89**	2.538	0.026	2.501	2.602	88.46	0.72	86.42	**88.89**
	34	7200.00	**91.36**	2.742	0.031	2.698	2.814	90.62	0.93	87.65	**91.36**
	35	7200.00	**95.06**	3.345	0.039	3.257	3.421	94.44	0.75	92.59	**95.06**
	39	7200.00	**90.12**	1.967	0.118	1.759	2.179	90.06	0.28	88.89	**90.12**
	42	7200.00	**80.25**	1.726	0.023	1.704	1.801	79.81	0.60	79.01	**80.25**
ISLAND											
 *10 × 10 (100 cells)* ***90*** *maritime (90.00 %)* *10 terrestrial (10.00 %)*	54	7200.00	**45.56**	1.694	0.047	1.626	1.782	45.33	0.46	44.44	**45.56**
	55	7200.00	**85.56**	2.205	0.020	2.176	2.260	84.94	0.67	83.33	**85.56**
	56	7200.00	98.89	2.895	0.023	2.850	2.934	99.11	0.58	97.78	**100.00***
	58	7200.00	**92.22**	2.722	0.032	2.674	2.794	91.50	0.83	90.00	**92.22**
	59	7200.00	**93.33**	2.693	0.025	2.645	2.770	92.61	0.75	91.11	**93.33**
	60	7200.00	**97.78**	3.149	0.047	3.070	3.249	96.83	0.41	96.67	**97.78**
	61	7200.00	**95.56**	2.103	0.023	2.076	2.152	95.44	0.34	94.44	**95.56**
	62	7200.00	**98.89**	2.607	0.033	2.556	2.678	98.39	0.57	97.78	**98.89**
RIVER											
 *22 × 22 (484 cells)* ***99*** *maritime (20.45 %)* *385 terrestrial (79.55 %)*	79	7200.00	**52.53**	1.571	0.045	1.497	1.647	51.92	0.69	50.51	**52.53**
	82	7200.00	**69.70**	1.915	0.042	1.850	2.010	68.54	0.59	67.68	**69.70**
	83	7200.00	**91.92**	2.089	0.028	2.048	2.143	90.81	0.73	88.89	**91.92**
	84	7200.00	**90.91**	2.010	0.031	1.969	2.080	89.75	0.75	87.88	**90.91**
	85	7200.00	**91.92**	2.188	0.030	2.148	2.243	91.11	0.62	89.90	**91.92**
	86	2057.99	**98.99***	2.573	0.022	2.539	2.622	97.68	0.66	96.97	**98.99***
	89	7200.00	**98.99**	3.382	0.038	3.310	3.467	97.07	1.18	94.95	**98.99**
	99	7200.00	**67.68**	1.584	0.020	1.553	1.637	65.91	1.39	62.63	**67.68**

Table 5. Numerical experiments with the best MILP from [12] and the Simulated Annealing (SA) metaheuristic developed in this paper on a selection of instances.

4.3 Example of Solution

A complete example of a solution obtained at the end of an execution of the SA on instance no. 58 (island group) is presented in Figure 5, which is broken down into several parts. **(a)** is the continuous heatmap of CDPs for the entire network over the whole AoI. **(b)** is the translation of the previous map in its discrete version, with the cells effectively covered (in red), i.e. where the point in the center is covered with a CDP greater than or equal to the detection treshold ($\phi = 0.95$). The white line highlights the separation between zones where the CDP is lower than the detection threshold and zones where it is higher. **(c)** is a bar chart that shows the importance of each buoy within the network. More precisely, this diagram quantifies the coverage rate that would be lost if a given buoy were to disappear (malfunction, destruction, etc.), in absolute values with the bar on the left and in relative values with the bar on the right. For example, if buoy no. 1 were to be removed, we would lose around 55 percentage points of coverage (absolute), which equates to a drop of around 60 % in the current coverage rate (relative). Note that this diagram is only valid for a single horizon and that it would have to be regenerated after the removal or addition of a buoy. **(d)** is a chord diagram representing the contribution of the different sonar systems made up by each of the buoys, adding complementary information to the previous diagram. Here, for a network $\omega \in \Omega$, the contribution of a given sonar system $(s, r) \in \Xi_\omega$ is calculated as follows: $\sum_{t \in T} \min\{\tilde{P}_d^{(s,r)}(t), 1\}$, with $\tilde{P}_d^{(s,r)}(t) = \log_{(1-\phi)}(1 - P_d^{(s,r)}(t))$. It is therefore the sum of the individual contributions, taking care not to include contributions greater than or equal to 1, because a target is considered to be covered as soon as the contributions are greater than 1. Hence, it is useless to know that a target is covered more than necessary by a given sonar system. Finally, in this diagram, the thicker the connection (the edge) between two buoys, the more significant the sonar system formed by these two buoys is within the network. **(e)** is a collection of heatmaps represented in matrix form with buoys having a source in rows and buoys having a receiver in columns. Thus, at the intersection of a row and a column we find the visualization of the individual contribution of a given sonar system. For example, at the intersection of row no. 1 and column no. 4 is the contribution of the sonar system formed by the source of the buoy TxRx with unique identifier 1 and the receiver of the buoy Rx with unique identifier 6.

5 Conclusions and Perspectives

In conclusion, in this paper, we have proposed a new method for efficiently solving the Area Coverage (AC) problem in Heterogeneous Multistatic Sonar Networks (HMSNs), based on a Simulated Annealing (SA) metaheuristic with a set of dedicated neighborhoods. We first take into account various factors including coastlines, probabilistic detection models, an adverse phenomenon referred to as direct blast and three categories of buoys (Tx, Rx, TxRx) with several types of buoys in each of these categories. Then, we also consider variable performance depending on the source/receiver pair as well as potential incompatibilities, for

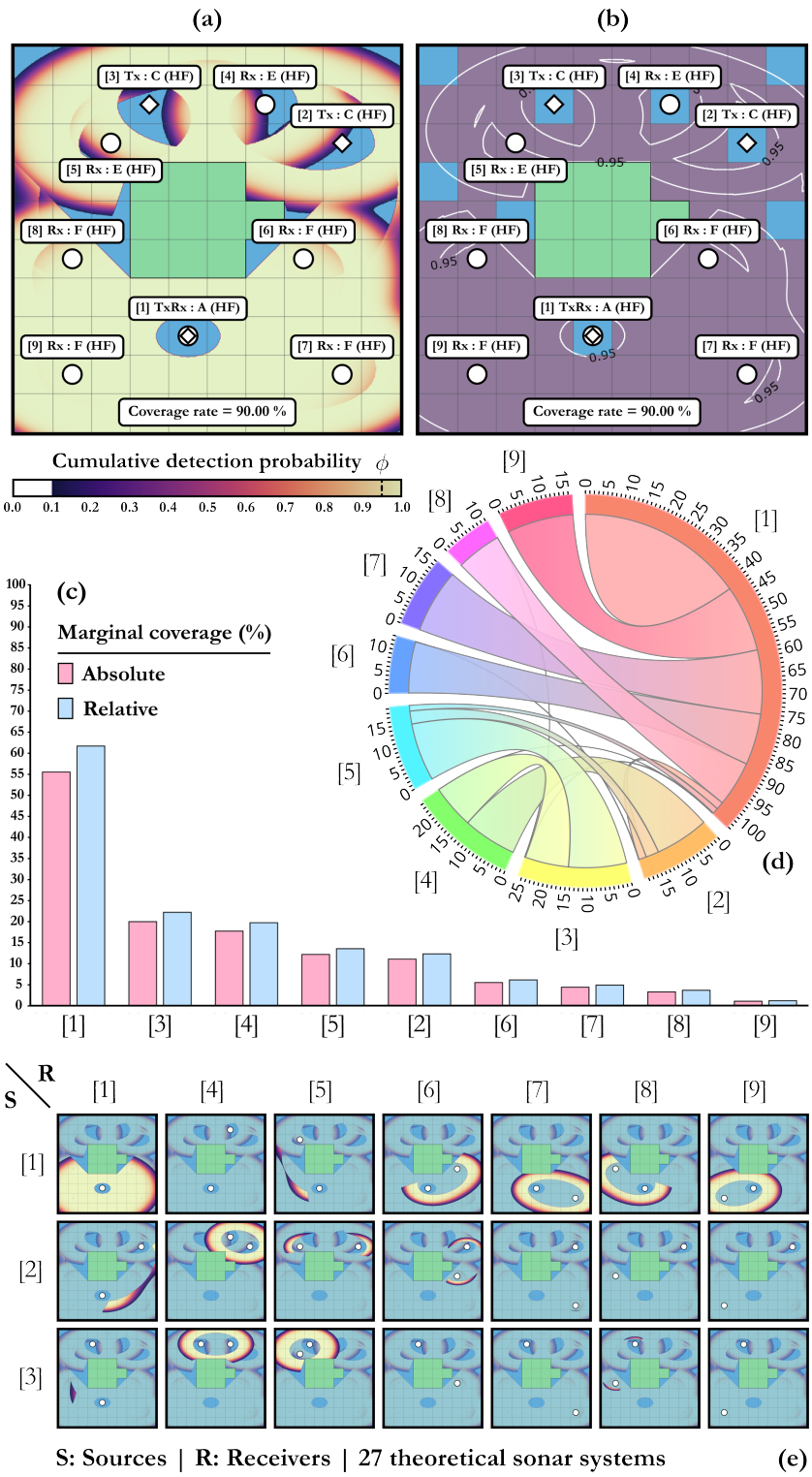


Fig. 5. Example of a Simulated Annealing (SA) solution on instance no. 58 (island group).

example because of different operating frequencies. As a result, we are able to process instances of interest from an operational point of view much more efficiently than current state-of-the-art methods. Furthermore, we introduce a novel and innovative framework designed to visualize the solutions we have derived. Future work could extend this method and these experiments to a larger collection of instances, taking care to further vary the sizes and and the geometric configuration of coastlines.

References

1. Church, R.L., Revelle, C.S.: The maximal covering location problem. Papers of the Regional Science Association **32**, 101–118 (1974)
2. Cox, H.: Fundamentals of Bistatic Active sonar. Springer, Dordrecht (1989). https://doi.org/10.1007/978-94-009-2289-1_1
3. Craparo, E.M., Karatas, M.: Optimal source placement for point coverage in active multistatic sonar networks. Naval Research Logistics **67**(1), 63–74 (Feb 2020). https://doi.org/10.1002/nav.21877
4. Franzin, A., Stützle, T.: Revisiting simulated annealing: A component-based analysis. Computers & Operations Research **104**, 191–206 (2019). https://doi.org/10.1016/j.cor.2018.12.015
5. Fügenschuh, A.R., Craparo, E.M., Karatas, M., Buttrey, S.E.: Solving multistatic sonar location problems with mixed-integer programming. Optimization and Engineering **21**(1), 273–303 (Mar 2020). https://doi.org/10.1007/s11081-019-09445-2
6. Golden, B.L., Skiscim, C.C.: Using simulated annealing to solve routing and location problems. Naval Research Logistics Quarterly **33**(2), 261–279 (1986). https://doi.org/10.1002/nav.3800330209
7. Holler, R., Horbach, A., McEachern, J., of Naval Research, U.S.O.: The Ears of Air ASW: A History of U.S. Navy Sonobuoys. Navmar Applied Sciences Corporation, Warminster, Pennsylvania (2008)
8. Khoufi, I., Minet, P., Laouiti, A., Mahfoudh, S.: Survey of deployment algorithms in wireless sensor networks: Coverage and connectivity issues and challenges. International Journal of Autonomous and Adaptive Communications Systems **10**, 341 (01 2017). https://doi.org/10.1504/IJAACS.2017.088774
9. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983). https://doi.org/10.1126/science.220.4598.671
10. Megiddo, N., Zemel, E., Hakimi, S.L.: The maximum coverage location problem. Siam Journal on Algebraic and Discrete Methods **4**, 253–261 (1983)
11. Thuillier, O., Le Josse, N., Olteanu, A.L., Sevaux, M., Tanguy, H.: Catalogue of coastal-based instances [data set] (2024). https://doi.org/10.5281/zenodo.10530247
12. Thuillier, O., Le Josse, N., Olteanu, A.L., Sevaux, M., Tanguy, H.: Efficient configuration of heterogeneous multistatic sonar networks: A mixed-integer linear programming approach. Computers & Operations Research **167**, 106637 (2024). https://doi.org/10.1016/j.cor.2024.106637
13. Thuillier, O., Le Josse, N., Olteanu, A.L., Sevaux, M., Tanguy, H.: An improved two-phase heuristic for active multistatic sonar network configuration. Expert Systems with Applications **238**, 121985 (2024). https://doi.org/10.1016/j.eswa.2023.121985
14. Urick, R.J.: Principles of underwater sound. McGraw-Hill, New York, 3rd edn. (1983)