Hybrid multi-objective node deployment for energy-coverage problem in mobile underwater wireless sensor networks

Salmah Fattah¹,², Ismail Ahmedy¹, Mohd Yamani Idna Idris¹ and Abdullah Gani²

Abstract
Underwater wireless sensor networks have grown considerably in recent years and now contribute substantially to ocean surveillance applications, marine monitoring and target detection. However, the existing deployment solutions struggle to address the deployment of mobile underwater sensor nodes as a stochastic system. The system faces internal and external environment problems that must be addressed for maximum coverage in the deployment region while minimizing energy consumption. In addition, the existing traditional approaches have limitations of improving simultaneously the objective function of network coverage and the dissipated energy in mobility, sensing and redundant coverage. The proposed solution introduced a hybrid adaptive multi-parent crossover genetic algorithm and fuzzy dominance-based decomposition approach by adapting the original non-dominated sorting genetic algorithm II. This study evaluated the solution to substantiate its efficacy, particularly regarding the nodes' coverage rate, energy consumption and the system's Pareto optimal metrics and execution time. The results and comparative analysis indicate that the Multi-Objective Optimisation Genetic Algorithm based on Adaptive Multi-Parent Crossover and Fuzzy Dominance (MOGA-AMPazy) is a better solution to the multi-objective sensor node deployment problem, outperforming the non-dominated sorting genetic algorithm II, SPEA2 and MOEA/D algorithms. Moreover, MOGA-AMPazy ensures maximum global convergence and has less computational complexity. Ultimately, the proposed solution enables the decision-maker or mission planners to monitor effectively the region of interest.

Keywords
Multi-objective optimization, mobile node deployment, metaheuristic algorithm, underwater wireless sensor networks, coverage, energy consumption

Date received: 31 March 2022; accepted: 6 August 2022
Handling Editor: Lyudmila Mihaylova

Introduction
Sensor nodes placed across a vast region in mobile underwater wireless sensor networks (UWSNs) can gather valuable data from the sensor field by communicating via wireless technology. Many wireless sensor systems, particularly remote monitoring and surveillance, can be placed in unstable regions, often without human intervention. However, optimal sensor deployment remains challenging due to the stochastic nature of the environment and the trade-offs between coverage and energy consumption. The existing approaches often struggle to address these issues simultaneously.

The proposed solution introduces a hybrid adaptive multi-parent crossover genetic algorithm and fuzzy dominance-based decomposition approach to improve the performance of multi-objective optimization in mobile underwater sensor networks. This study evaluates the proposed algorithm's efficacy in terms of coverage rate, energy consumption, and Pareto optimal solutions. The results show that the Multi-Objective Optimisation Genetic Algorithm based on Adaptive Multi-Parent Crossover and Fuzzy Dominance (MOGA-AMPazy) outperforms traditional algorithms like non-dominated sorting genetic algorithm II (NSGA-II), SPEA2, and MOEA/D. MOGA-AMPazy ensures maximum global convergence while maintaining a lower computational complexity, making it a promising solution for mobile underwater sensor network deployment.
placement is the most significant utilization of available sensors. A sensor must be placed in a contextually suitable position before it can provide valuable data to the system. Node deployment is, therefore, a crucial process in underwater sensor networks. It supports many essential functions, such as routing protocol, localization and network architecture. Moreover, node deployment significantly affects network performance. Sensor nodes in mobile deployments may operate freely in any orientation.

The existing mobile UWSN solutions have enormous power to maximize network coverage. However, underwater mobile sensor node deployment has several issues. For instance, arbitrary movement of underwater mobile nodes or vehicles can cause swift and unpredictable changes to the network topology. Whenever a node moves out of place, the network topology must be re-calculated and re-distributed. Based on the current knowledge about the target, an underwater mobile sensor may adjust the initial node location. The sensor node location is then rearranged to achieve a standard end configuration. The network may also experience redundancy or coverage holes because of sensor failure, destruction or unusual events. Therefore, nodes require immense energy to communicate with the deployed sensors. In a centralized approach, every sensor node must send its position information to a control centre. The control centre performs computations based on global information about the network topology before instructing the nodes about subsequent directions and movement. An increase of mobile sensor nodes might overload the control centre, affecting processing time. In addition, UWSNs generally have a long propagation latency and high bit error rate. The long propagation may lead to the central controller making bad node position movement choices in highly dynamic situations.

Hence, designing a decentralized system that uses local information to make movement decisions is necessary. The limited communication bandwidth of mobile nodes also inhibits network performance in the underwater environment, creating a potential bottleneck for network expansion. Mobile node allocation to a UWSN environment faces several conflicting objectives. In this study, the sensor node deployment tasks were required to satisfy maximum coverage in the deployment region while minimizing energy consumption. Considering all objectives when solving one problem is undoubtedly challenging and requires effective methods.

This study resolves the abovementioned issues by offering a meta heuristic optimization-based deployment approach for mobile underwater sensor nodes. The proposed solution aims to improve network coverage and reduce energy consumption and execution time with a multi-objective optimization (MOO) genetic algorithm (GA) based on the adaptive multi-parent (AMP) crossover and fuzzy dominance.

**Objectives**

This study aims to enhance the mobile underwater sensor node deployment model by presenting a hybrid multi-objective formulation. The following set of objectives must be achieved to fulfill the research aim:

1. To analyse several mobile underwater node deployment algorithms by evaluating coverage rate, node energy consumption and execution time.
2. To develop a hybrid fuzzy dominance-based decomposition technique and AMP crossover GA for mobile UWSNs to optimize conflicting deployment objectives.
3. To evaluate the proposed solution and to validate and compare its performance with that of other existing techniques.

**Motivation**

This section describes the motivation for mobile sensor node deployment concerning UWSN technologies. Some regions are particularly vulnerable to border encroachments and traditional threats such as kidnappings, illegal immigration and smuggling. Regional authorities face a number of challenges in monitoring such locations. Therefore, this study proposes mobile sensors in UWSNs as an alternative solution to assist authorities in monitoring the region of interest. Many undiscovered resources – especially seas, sensors and sensor networks – warrant continuous study and investigation. In particular, UWSN theories and applications require further study and evaluation, given that UWSNs have shown substantial market growth worldwide. However, a wide variety of application needs have imposed critical limitations on network job performance and sensor node capabilities in the monitored area; research works have paid minimal attention to deploying multiple mobile sensor nodes in UWSNs with multiple-objective solutions.

**The contribution of this work**

The contributions of this study are as follows:

1. Review state-of-the-art existing mobile sensor nodes deployment solutions in UWSNs.
2. A MOO GA was developed based on AMP and fuzzy dominance by adapting prospect theory to deploy mobile sensor nodes in UWSNs.
3. The coverage rates, energy consumption, Pareto optimal metrics and execution time of the
The proposed solution were evaluated against existing approaches.

The remainder of this paper is organized as follows. Section “Related works” analyses the state-of-the-art mobile sensor node deployment techniques in UWSNs. Section “Proposed solution” describes the MOO algorithms and components of the proposed solution. Section “Results and discussions” presents the evaluation of the results and compares them with existing mobile sensor nodes deployment approaches. Finally, section “Conclusion and future works” provides concluding remarks and future directions for related research.

Related works

This section discusses recent studies of mobile underwater sensor nodes deployment solutions by carefully examining each work and reporting on them separately. UWSN deployment algorithms can be categorized into random, planned, incremental and movement-assisted. In movement-assisted deployment algorithms, sensors are deployed randomly and then moved using the knowledge of other node locations. The major movement-assisted or mobile approaches, among others, are computational geometry (CG), artificial potential field and metaheuristic algorithms.

CG research has aimed to develop practical deployment algorithms for static and mobile UWSNs. Wang et al.3 proposed the double-coverage algorithm to solve the premature failure caused by excessive energy consumption in UWSN sensor node deployment by focusing on deploying mobile nodes in UWSNs. Four mobile sensor nodes were mounted on the four vertices of the rectangle to double-cover the rectangular area. However, the model only covers 58% of the sensing area. Furthermore, Shen et al.4 developed a fully distributed deployment algorithm to automatically transfer all mobile sensors distributed in underwater space to desired locations, resulting in maximum-level underwater high k-barrier coverage. The proposed algorithm outperforms the Hungarian and HungarianK approaches regarding the length of construction of k-barrier coverage. However, the study did not consider energy consumption when evaluating the algorithm’s efficiency. Dang et al.5 suggested a target detection coverage algorithm based on three-dimensional (3D) Voronoi partitioning (3D-VPCA) to ensure the network’s overall reliability. The study extended Voronoi division based on the two-dimensional plane, allowing 3D-VPCA of sensor nodes in 3D regions. The findings revealed that the 3D-VPCA algorithm could substantially reduce network energy consumption while increasing the coverage ratio. Nonetheless, as one of the success factors for energy efficiency, the mobility of sensor nodes must be calculated.

The study proposed an original real-time obstacle avoidance strategy for mobile robots based on ‘an artificial potential principle’. A mobile robot is thought to be travelling through a region of artificial forces or virtual forces. The oceans are complex fluids stratified and rotating.6 To resolve these issues, Mahboubi and Aghdam7 and Fang et al.8 presented a node mobility strategy to optimize sensor coverage in each 2D sensing area based on the virtual force model and Voronoi diagram. The authors applied virtual forces to each sensor from the vertices and boundaries of the Voronoi cell that drives it. Simulations showed that the proposed algorithms could successfully increase sensing coverage. However, as the network models did not consider the effect of water flows on network coverage, the method cannot be extended explicitly to marine environments. Liu et al.9 expanded the 2D virtual force model to 3D environments. The authors proposed a distributed node deployment algorithm based on virtual forces (DABVF) to increase the network coverage of UWSNs by analysing the significance of anchor nodes in a marine environment. The DABVF increased network coverage, lowered node energy consumption, balanced residual energy in nodes and optimized node distribution. However, the proposed algorithm did not provide network topology repair strategies for sensor node failure.

Metaheuristic algorithms can solve complex optimization problems effectively and efficiently. They are either trajectory-based or population-based from one perspective. A recent study by Yakica and Karatas10 concluded that the metaheuristic approach could create diverse and optimum solutions with less execution time. Zou et al.11 defined a particle swarm optimization (PSO)-based topology control mechanism for autonomous underwater vehicles (AUVs) operating in unknown 3D underwater spaces, dubbed 3D-PSO. The study’s simulation results show that the algorithm can quickly and efficiently spread AUVs apart and completely secure naval assets for 3D underwater missions. However, these approaches cannot be used to change the locations of sensors in complex environments with uncertain events to ensure the desired monitoring efficiency. As previously reported by Harizan and Kuila,12 metaheuristic approach can produce better and more effective results than other algorithms, especially for solving coverage and connectivity problems. A more comprehensive description of nature-inspired metaheuristic can be found by Singh et al.13 The discussion covers the implementation of optimisation algorithms and their performance in solving problems in sensor networks.
To solve these issues, Du et al.\textsuperscript{14} investigated the problem of 3D underwater sensor deployment with non-uniform coverage requirements. The authors used the Lagrange flow model in the design and simulation of PSSD and evaluated the model using a theoretical knowledge metric. The dynamic simulation experiment results revealed that the sensors spent more time rotating to cover the events, resulting in more energy usage. Another study suggested a method to determine enough active nodes in UWSNs at various times to cover the targets that must be detected.\textsuperscript{15} The proposed multi-population harmony search algorithm considers the scheduling issue of dynamic heterogeneous UWSNs in 3D space. This algorithm uses a sleep scheduling scheme to manage sensor operations, in which sensors take turns working or sleeping to avoid redundant power consumption or malfunction. Developing a system that can change sensor positions based on environments and objectives is critical for optimizing target coverage monitoring. Wang et al.\textsuperscript{16} proposed a distributed hybrid fish swarm optimization algorithm based on the impact of water flow and the activity of an artificial fish swarm system to increase the coverage efficacy of event collection and prevent blind sensor node movements. Zhang et al.\textsuperscript{17} also proposed a new optimal coverage enhancing deployment approach for 3D underwater sensor networks based on an improved fruit fly optimization algorithm, which aimed to determine the optimization strategy with the fewest iterations and the most significant objective function. However, even if the algorithm has the power to escape from the local optimum, it also may collapse into it. Lv and Li\textsuperscript{18} suggested using a probabilistic sensing-based algorithm for improved sensor network k-coverage. Maximum weight matching was used to implement a centralized allocation strategy that effectively reduces the energy consumed during node allocation. Random cluster deployment of sensors was used in the proposed work.\textsuperscript{19}

Wang et al.\textsuperscript{20} recently conducted a study to place the sensor node using a multi-objective algorithm to maximize sensing coverage and minimize the distance between the sensor and the target point. However, the study did not consider the energy consumption of sensor nodes and was not implemented in an underwater environment. Apart from that, Jebi and Baulkani\textsuperscript{21} also propose a multi-objective algorithm to prolong the connectivity and coverage in a wireless sensor network. However, the proposed method has a high delay. The delay can cause communication and transmission between nodes to be interrupted. Overall, none of the algorithms mentioned above considers the coverage rate of network changes and node energy consumption. Maximizing coverage means maximizing the space occupied by the mobile sensor nodes concerning the size of the total deployment space. These are innately conflicting objectives. Table 1 summarizes the techniques for deploying mobile sensor nodes and the related studies’ primary objectives, control strategies, design factors, mobility models and application of the associated works.

**Proposed solution**

The model proposed in this study involves conflicting optimization problems (MOO problems) that simultaneously minimize or maximize objective functions. The model development procedure involved four main phases: model development, objective formalization, solution design and evaluation. Figure 1 shows the methodology of the research. The first phase involved constructing four main models for the network, its coverage and energy consumption and MOO. The network model establishes the environment for underwater mobile sensor nodes. Each node executes the algorithm without using the central node. This study assumed that all mobile sensor nodes are identical or homogeneous, particularly in computational ability.

The coverage model includes the sensing model, coverage ratio calculation and coverage algorithm implementation. The energy consumption model addresses a particular focus of this study: the network’s energy consumption owing to mobility, sensing and data communication. The next phase comprised the objective optimization formulation, which is the focus of this study. Figure 2 depicts the model diagram, highlighting its inputs, algorithm and outputs. The main contribution of this study’s optimization process is a hybrid solution involving a multi-objective genetic algorithm (MOGA), AMP, fuzzy dominance and prospect theory.

The final phase of the study evaluates the proposed solution. At the end of this phase, the analysis includes coverage rates, energy consumption rates, Pareto metrics and execution time using specific tools. The solution is also compared with the existing methods’ results to determine the effectiveness of the proposed solution.

**Experimental setup**

This research considers several simulations to evaluate the output by checking the validity of the proposed algorithm. The study conducted the simulations using a computer with a Processor 2.1 GHz-CoreTM i7 CPU, 64-bit Windows 10 operating system, 1 Mb cache and 8 GB RAM using MATLAB R2018b. The object-oriented environment in MATLAB can keep private meta-data and combine activities with agents. The system environment consists of a sensor, communicator, node and network classes. The sensor contains a string identification, a value gathered from the environment and essential functions to examine new data. This class
Table 1. Comparison of state-of-the-art mobile sensor nodes deployment solutions in UWSNs.

| Paper reference | Deployment techniques | Deployment objectives | Control strategy | Design factors | Mobility model | Applications |
|-----------------|-----------------------|-----------------------|------------------|----------------|----------------|--------------|
| MCMC            | CG                    | APF                   | APF              | MCMC           | Other          | Environmental monitoring |
| Double-coverage | CG                    | APF                   | APF              | MCMC           | Other          | Target monitoring |
| FDDA            | CG                    | APF                   | APF              | MCMC           | Other          | Intrusion detection |
| 3D-VPCA         | CG                    | APF                   | APF              | MCMC           | Other          | Maritime surveillance |
| 3D-PSO          | CG                    | APF                   | APF              | MCMC           | Other          | Environmental monitoring |
| UFOA            | CG                    | APF                   | APF              | MCMC           | Other          | Target monitoring |
| Probabilistic   | CG                    | APF                   | APF              | MCMC           | Other          | Water resource monitoring |
| Improved        | CG                    | APF                   | APF              | MCMC           | Other          | Environmental monitoring |
| k-coverage      | CG                    | APF                   | APF              | MCMC           | Other          | Target monitoring |
| DABVF           | CG                    | APF                   | APF              | MCMC           | Other          | Event monitoring |
| PSSD            | CG                    | APF                   | APF              | MCMC           | Other          | Target monitoring |
| MPHSA           | CG                    | APF                   | APF              | MCMC           | Other          | Event monitoring |
| DHFSOA          | CG                    | APF                   | APF              | MCMC           | Other          | Event monitoring |

CG: computational geometry; APF: artificial potential field; MCMC: markov chain monte carlo; FDDA: fully distributed deployment algorithm; 3D-VPCA: three dimensional voronoi partitioning coverage algorithm; 3D-PSO: three dimensional particle swarm optimization; UFOA: fruit fly optimization algorithm; DABVF: distributed node deployment algorithm; PSSD: particle swarm-inspired underwater sensor deployment; MPHSA: multi-population harmony search algorithm; DHFSOA: distributed hybrid fish swarm optimization.
can emulate a given sensor’s behaviour, consisting of the power specifications and configuration to provide data.

The sensor class contains an identification string, a value gathered from the environment and essential functions to examine new data. To provide data, this class can emulate a given sensor’s behaviour, including the power specifications and configuration. The communicator class provides a basic implementation of a communication modem. The communicator consists of send and receives queues, power requirements and success rates, which serve as an intermediary for the type of device and medium used for transfer. The node class operates as a container of communicators and sensors. Nodes store information concerning their position, neighbours’ list and location, identification value and three-dimensional coordinates. The algorithm is executed within each node, including all essential support functions for decision-making. The nodes further provide information concerning sent messages and power consumed for external classes and scripts. The network class of this system acts as a container for all other containers. It contains the nodes, supports a periodically varying connectivity matrix and list of positions and
stores simulation metadata. The network class also holds data on all sent and received messages, power consumption, distance travelled and algorithm execution time per node.

A network stores nodes’ locations and communication paths based on location and range information. The system provides an ID, area and neighbours’ list as nodes are formed. The nodes reference the MATLAB code’s version, which runs in an environmental implementation. Its implementation strengths are flexibility, ease of continuing functionality, enhanced maintenance and efficiency achieved by scaling down complete references. A series of scripts and helper machine functions was set up such that non-damaging changes simplify the calculations. The scripts and operations can be divided into three categories: initializers, modifiers and run-time. Initializers initialize a data field (e.g. node positions, message queues) and determine constants for the simulation. Simultaneously, modifiers produce temporary partial copies of matrices and return the results. Modifiers are helper functions and scripts that handle storage systems. Last, the run-time operation runs the simulation – the process of generating messages, measuring how to respond and organizing all run-time tasks – with algorithmic helpers. During the experiment, the nodes moved randomly to resemble the dynamic existence of the real world.

Algorithm description

This section discusses the algorithms used to develop the proposed solution.

Coverage rate algorithm. A sensing model initially determined the coverage of UWSNs. Two types of sensing models are typically used for simulating sensor performance: binary and probability models. This research focuses only on the probability model because it can calculate the chance of a target occurring in each region with a value that varies between 0 and 1. If a sensor node S is present at location \((x_a, y_a, z_a)\), then the sensing range of S is assumed to have radius \(R_s\) centred at \((x_s, y_s, z_s)\). The sensing radius of node S is \(R_s\). The distance between the target and sensor in the three-dimensional area can be calculated as in equation (1)

\[
j_s = \sqrt{(x_a - x_s)^2 + (y_a - y_s)^2 + (z_a - z_s)^2}
\]  

A sensor has two critical distances in the probability model. The first is \(R_j\), which is the same as the binary model’s version. The sensor can identify the target with a probability of 1 if the distance between the target and the sensor is smaller than \(R_j\). The second distance is the ambiguous range, denoted by \(R_k\). If the distance between the target and the sensor is between \(R_j\) and \(R_k\), then the likelihood that the sensor will detect the target is proportional to the distance.

The sensor will not recognize the target if its distance is greater than \(R_j + R_k\). The probability model is expressed as in equation (2)

\[
P_m = \begin{cases} 
0, & \text{if } (R_j + R_k < d_s) \\
\exp(-\lambda d^2), & \text{if } (R_j < d_s \leq R_j + R_k) \\
1, & \text{if } (d_s \leq R_j + R_k)
\end{cases}
\]  

Figure 3. Sensing range of sensor nodes.

\(R_j + R_k\) (Figure 3), then the probability model provides a steadily declining probability. Both sensing models will offer a detection probability of 0 when the distance is more significant than \(R_j + R_k\). When \(R_k\) is zero, the probability model is more realistic than the binary model. The binary model is a simplified version of the probability model.

As shown in equation (3), the coverage ratio \((R_{Cov})\) is the ratio of the area that can be covered cooperatively by sensors \((A_{Cov})\) to the full sensing field \((A_{Sum})\)

\[
R_{Cov} = \frac{A_{Cov}}{A_{Sum}}
\]  

The maximum coverage ratio is 1. When \(R_{Cov}\) is 1, the sensors entirely cover the space or area. If at least one sensor node – or the combined detection of several sensors – can cover an area, it is considered covered. If the entire sensing field has \(N\) sensor nodes, the joint detection probability for a specific area can be calculated as in equation (4)

\[
P_{joint} = 1 - \prod_{s=1}^{N} (1 - P_m)
\]
$P_{\text{joint}}$ can be any value between 0 and 1 in the probability model. A threshold ($P_r$) is necessary to determine if an area is covered. The area is covered if $P_{\text{joint}}$ is more significant than $P_r$; the region is not covered. The application’s specifications determine the value of $P_r$. As sensors are placed randomly, the region they cover will be uneven.

The nodes in the network should meet the sensing probability $P_{\text{joint}} \geq P_r$ of the grid points in the sensing field more than feasible during the deployment and coverage operations of the sensor nodes to achieve full coverage. Figure 4 shows the algorithm description for the node deployment method to maximize the coverage of the observation area according to the sensing rate.

When the network complies with a coverage requirement of k-coverage or when the deployed sensor nodes already approach the maximum value of the set-up, the algorithm stops working. Every iteration of an algorithm completes the deployment of the sensor node in the monitoring region. Each iterative procedure deploys a single sensor node at a specified grid point.

**Energy consumption algorithm.** The MOGA and fuzzy dominance-based decomposition (MOGA-AMPazy) algorithm considers the energy consumed for mobility, sensing and data communication processes. Its main objective is to improve the coverage area by relocating the underwater mobile sensor nodes. Each sensor node uses energy as it moves or turns throughout the relocation process. This study considered the power utilized for moving to be constant when the weight of sensors and speed are both constant. The moving distance $D_{\text{mov}}(i)$ of the sensor $s_i$ equals to the distance between the initial and final locations in the suggested solution because each sensor moves just once. Energy is also consumed when a sensor node rotates.

The amount of energy used to turn ($E_{\text{turn}}$) is proportional to the angle to which the sensor node is rotated. A sensor can rotate clockwise or counter clockwise and choose the results in a smaller spinning angle towards the direction it must go

$$D_{\text{movSum}} = \sum_{i=1}^{n} D_{\text{mov}}(i) = \sum_{i=1}^{n} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}$$

$$T_{\text{sum}} = \sum_{i=1}^{n} \Delta T_i$$

$D_{\text{movSum}}$ in equation (5) and $T_{\text{sum}}$ in equation (6) represent the total distance travelled and the total angle turned by the sensor, respectively. When acceleration is slight, the energy required for sensor movement $s_i$ is linear concerning the distance travelled.\(^{22}\) This notion is represented in equations (7) and (8)

$$E_{\text{mov}} = K_{\text{mov}} \times D_{\text{movSum}}$$

$$E_{\text{turn}} = K_{\text{turn}} \times T_{\text{sum}}$$

where $K_{\text{mov}}$ and $K_{\text{turn}}$ are coefficients that describe the energy consumption rate. Reshko et al.\(^{23}\) developed omnidirectional mini-robots that can be employed in mobile sensors. Each robot in their study had three uniform wheels around the edge of a round-shaped platform placed 120° apart. The wheels were arranged in a rectangle. The robot could then go straight ahead or turn in any orientation.

Mei et al.\(^{24}\) offered an energy model for the omnidirectional mobile robot. According to their research, the energy required for a robot platform to move 1 m at a constant speed of 0.08 m/s is 9.34 Joules. This three-wheel robot required 2.35 Joules of energy to turn 90°. The sensors in the algorithm were assumed to travel and turn at a consistent rate. As a result, the parameters provided by equations (9) and (10) can be used to define $K_{\text{mov}}$ and $K_{\text{turn}}$

$$K_{\text{mov}} = 9.34 \text{ (J/m)}$$

$$K_{\text{turn}} = \frac{2.35}{90} = 0.0261 \text{ (J/m)}$$

The total energy required to move the sensors will be the sum of the two types of energy (equation (11))

$$E_{\text{Mobility}} = E_{\text{mov}} + E_{\text{turn}}$$

Range customisable sensors were employed because they consume less power when reduced sensing radius. Different energy models were used to study the relationship between a sensor’s sensing radius and its energy consumption depending on the sensing device’s properties. The linear, quadratic, cubical and quadruplicate models are standard. The most used are the linear and quadratic models by Cardei et al.\(^{25}\) and Dhawan et al.\(^{26}\)
The quadratic model is used in this work, in which the sensor ($s_i$) energy consumption is proportional to the sensing radius, as indicated in equation (12)

$$E_{\text{Sensing}} = K_{\text{sense}} \times R_s^2 (R_s \leq R_{\text{max}})$$

where $E_{\text{Sensing}}$ is the energy in sensing and $K_{\text{sense}}$ is a constant representing the energy consumption rate. Sensor nodes use considerable energy delivering and receiving data during the data process. The nodes not performing tasks enter an idle or sleep state at this stage. The energy level of the sensor nodes changes with each cycle of the data transmission process because of the quantity of energy expended throughout the data transmission cycle. The energy the nodes require for sending data contributes to energy consumption. The data transmission range, time and minimum power packets that can be received must be considered. Energy consumption is calculated as in equation (13)

$$E_{\text{DC}} = D_t \times P_{\text{min}} \times T(d)$$

where $d$ denotes data transmission distance, $D_t$ indicates data transmission time and $P_{\text{min}}$ denotes the lowest power packets that can be received. The underwater acoustic signal attenuation model calculates the transmission distance $T(d)$. As a result, equation (14) is used to compute the total energy consumption of each mobile sensor node ($E$)

$$E = E_{\text{Mobility}} + E_{\text{Sensing}} + E_{\text{DC}}$$

**Differential fuzzy dominance (DFD) algorithm.** The DFD function must be combined with the following method to identify the locations of nodes with high event detection. The method uses the Pareto front from a multi-objective representation to choose the optimal solution for high event detection in deployed node areas. This method selects the best individual from the ranked population (i.e. with the lowest ranking value) and adds it conditionally to the alpha set. Each solution may be allocated a single, smooth dominance to represent the degree of domination of the solution after establishing its membership role. A solution with a lower fuzzy dominance value is a better approach. Thus, sorting alternatives according to their fuzzy dominance values may assist in rapidly locating solutions around the Pareto front. K solutions were found and selected for each solution generation round using a fuzzy dominance sorting method. Figure 5 shows the algorithm.

**AMP algorithm.** AMP mainly aims to produce new offspring to ensure consistency between exploration and exploitation. If the offspring are far from their parents, they could be too diverse and require a greater convergence interval. Premature convergence may occur if the offspring’s distribution is less than that of their parents. This study presents an AMP-integrated GA (Figure 6), in which the number of parents participating in the crossover is varied according to the new generation’s performance. The number of parents is increased or decreased depending on the current generation’s performance relative to the preceding generation. This strategy guarantees that children are neither too different from nor like their parents. Control parameters are managed through adaptive GAs in various ways, including changing mutation probabilities over time, moving mutation probabilities between two fitness-based values and manipulating genetic operators. The proportion of parents changes because of this study’s findings, and substantial exploration of the problem space has resulted from these adaptive techniques. AMP produces offspring more likely to go to a more significant part of the search space, but others are less likely to go to more diverse points.

**Prospect theory algorithm.** Tversky and Kahneman proposed prospect theory to model human decision-making in the face of risk. Their approach applies to subjective probability and enables decision-making based on context and loss aversion in the present condition. The MOGA-AMPazy method uses prospect theory to establish stability amid exploration and exploitation dynamically. This study can solve multi-
objective problems with reasonable computational requirements with parallel dynamism based on probability weighting. This study uses prospect theory to make decisions based on success probabilities and risk evaluation.

Prospect theory takes a descriptive approach to model how humans make risk-informed decisions. The theory suggests that risky decision-making can be viewed as a selection among many gambles or possibilities. It makes an outstanding contribution as a descriptive decision-making model that divides decision-making into editing and evaluation processes. Editing is the first stage, in which individuals collect and process information using a ‘framework’ and ‘reference point’. In this study, the reference point \( r_0 \) is defined to determine the individual’s return and loss in the portfolio, representing the zero return or zero loss that the individual thinks. Evaluation is the second stage, in which the value and weight functions concerning subjective probability evaluate information. This study depended on the subjective probability weight function \( \pi(p) \) and the value function to evaluate information. The results were allocated to gains and losses in prospect theory rather than tangible assets; the value function reflected relative gains or losses and was defined as a power function (equation (15)) by Tversky and Kahneman\(^ {27} \)

\[
v(r) = \begin{cases} 
  p^\alpha, & r \geq 0 \\
  -\gamma(-r)^\beta, & r < 0
\end{cases} \quad (15)
\]

where \( \alpha \) and \( \beta > 0 \) measure the curvature of the value function for gains and losses, respectively, and \( \gamma \) is the loss aversion coefficient. Thus, the value function for gains (losses) is increasingly concave (convex) for smaller values of \( \alpha(\beta) > 0 \), and loss aversion is more pronounced for larger values of \( \gamma > 1 \).

Prospect theory assumes that individuals do not weigh outcomes by probability but by some distortion of probabilities. Prospect theory’s probability weighting function captures this distortion of probability. Two evident reference points exist for the probability: certainty and impossibility, or a 100\% chance and a 0\% chance. The weighting function can be parameterised in the following form (equation (16)) according to the probability weighting function proposed initially by Tversky and Kahneman\(^ {27} \)

\[
\pi(p) = \frac{p^\alpha}{(p^\alpha + (1 - p)^\gamma)^1/\gamma} \quad (16)
\]

where \( p \) is the weighting probability of the distribution of gains or losses, and \( >0 \) measures its degree of curvature. Therefore, the prospect theoretical utility function \( \pi(p) \) and \( v(r) \) can be represented in equation (17)

\[
PT_U = \sum_{s=1}^{S} \pi(p_s)v(r_s) \quad (17)
\]

The prospect theoretical weight function \( \pi(p) \) is used to measure the size of the event’s desirability to the result. \( \pi(p) \) is an increasing function, \( \pi(0) = 0, \pi(1) = 1 \), when the probability \( p \) is very small, that is \( \pi(p) < p \). The prospect theoretical value function \( v(r) \) represents the behavioural value of the obtained or lost result. For a given reference point \( r_0 \), as the function reflects the attitudes of different investors on the gains and losses, the parameters are different. Figure 7 shows a summary of the algorithm.

**Algorithm objective and fitness function design.** For this study, the sensor node deployment task was formulated as an unconstrained MOO problem to find a sensor node deployment arrangement in the 3D space that satisfied the required objectives. The MOGA-AMPazy solution has two objectives:

1. To maximize the coverage ratio, defined in equation (3).
2. To minimize the sensor energy used for mobility, sensing and data communication.

In MOGA-AMPazy, the fitness function was designed according to the proposed objectives. The optimum fitness function provides the optimal Pareto front (i.e. the best compromise between objectives) in optimization issues with many conflicting objectives, such as minimizing energy and maximizing the coverage ratio. In this research, the final sensor positions needed to be optimized. Therefore, sensor node coordinates were used as input variables for the GA.

The method is implemented to produce the fitness function’s minimal value. Thus, maximizing the coverage ratio involves minimizing the objective function using the uncovered area ratio (equation (18))

\[
R_{\text{unCov}} = 1 - R_{\text{Cov}} \quad (18)
\]
where $R_{Cor}$ is calculated using equation (19). The goal of the MOGA-AMPazy solution is to reduce the average travelling distance and the ratio of uncovered to covered area. Notably, a trade-off exists between the two goals. The sensors stay in their placements after the $N$ sensors have been deployed before the algorithm is run. The longer the sensors must move, the greater the coverage required.

Conversely, such long movements of the sensors do not guarantee broader coverage. A MOGA can achieve the best trade-off between the coverage and the travelled distance. Therefore, this study defines the fitness function of coverage as

$$\text{Minimize } F_1 = \sum_{i} R_{\text{runCov}(i)}$$

(19)

The second objective – to decrease the amount of energy used by sensors – was addressed using the method suggested by this research, which focuses on mobility, sensor modules and data transmission energy. The method does not expand to the number of computation sensor nodes required, so little communication overhead is produced. This algorithm only requires sensor movement after deployment. Hence, the energy consumed during sensor movement is a short-term consideration. However, the energy needed to move the sensor is much larger than long-term sensing energy consumption. Energy consumption must be minimized to maximize sensor lifetime. The total energy consumed by the sensors is obtained from equation (20). Relocation optimization is required in a UWSN if any sensors fail. Therefore, this study defines the energy consumption fitness function as follows

$$\text{Minimize } F_2 = E_{\text{total}} = \sum_{i} E_i$$

(20)

**MOO algorithm**

In this study, the difficulty of establishing the objective function was considered within the MOO algorithm’s solution. A collection of optimum locations (known as the Pareto front) corresponds to a particular trade-off between the values of the objective functions. The Pareto optimal point for the ‘best trade-off’ must be selected. This study used a GA to do so because such algorithms are well adapted to MOO problems being based on biological processes that are innately multi-objective. For commercial and military applications that demand rapid, adaptive, automated and self-learning procedures, GAs facilitate the discovery of approximation solutions. A GA uses three approaches based on selection techniques and fitness functions – criteria selection, aggregation selection and Pareto selection. In this research, Pareto selection was characterized by the non-dominated sorting genetic algorithm (NSGA-II). The primary concept behind this research was to establish a MOGA-based computational model for a mobile UWSN optimization principle that produces an adequate trade-off when addressing a surveillance monitoring application. A hybrid fuzzy dominance-based decomposition technique and AMP GA were applied to the original NSGA-II. The method compared two solutions using a fuzzy Pareto dominance and employed the scalar decomposition approach when one of the alternatives failed to dominate the other on a fuzzy dominance level.

A mobile sensor node deployment solution allocates available nodes in an underwater surveillance network to target surveillance. Given a known target, the initial node position and velocity must be chosen, and surveillance regions assigned such that a given set of objectives is optimized. The detecting range of the sensors is $r$ if the mobile sensor nodes are dispersed randomly over the region of interest. The proposed GA begins with a random population of individuals (the distribution of the initial nodes). The objective function then analyses the constraints’ satisfaction level in each iteration. Each repetition of the algorithm improves the solution (population). The operators oversee this improvement (crossover and mutation). A halting condition is used to bring the algorithm to a halt. The algorithm initializes the length and width of the region, the number of mobile sensor nodes, the number of targets, the chromosomes and the random locations of the chromosomes.

The mobile sensor nodes can adapt to changing situations, such as addition, loss or dysfunction, using a 3D evolutionary algorithm to find chromosomes with a minor fitness function and the fastest speed and movement directions. Each node informs its neighbours of its position and number of neighbours. The number of near neighbours within a node’s communication range (denoted as $D_i$) determines its degree $N_p$. Each node calculates its next optimal location and speed based on this local knowledge from its near neighbours. The fitness function is a crucial component that adjusts movement speed and direction based on the node’s current velocity vector and its neighbours’ relative locations, resulting in a better future position, if possible. The new value of the velocity vector is determined by the number and distance of neighbouring nodes within its communication range. A lower fitness score indicates that a node is in a better position. The movable sensor nodes will only travel to a new position with a higher fitness rating; otherwise, the vehicle will stay.

The fuzzy dominance process is then implemented by combining the Q solution with parent population P from the previous iteration. Later, the proposed
algorithm sorts the combined population \((P + Q)\) and executes the Q solution selection. If the set criteria of system termination are achieved, the system will be terminated. Otherwise, the AMP balances exploration and exploitation by providing new offspring. AMP manipulates control parameters by tweaking genetic operators over time, changing the mutation probability between two values depending on the fitness value. Thus, the number of parents changes depending on the fitness rating. The AMP algorithm was executed until there was no overlap in the solution search.

Result and discussion

This section describes the study’s data analysis and presents the outcome of a comparative analysis with existing algorithms. The proposed solution was critically evaluated through various tests and techniques. This section compares the proposed solution’s effectiveness for coverage rates, energy consumption, Pareto optimal metrics and execution time with existing algorithms.

Performance analysis of coverage rates

This study conducted two case studies to evaluate the proposed solution’s performance against three other existing algorithms. This study used the experiment’s collected data to analyse and compare the suggested solution’s coverage rate against the existing algorithms. The coverage ratios were obtained using the maximum travelled distances of all sensors in mobility for all tests. Compared with NSGA-II, SPEA2 and MOEA/D, MOGA-AMPazy exhibited improved coverage rates and travelled distance. The increase in coverage rate was minimal (Figure 8); however, the increment in the travelled distance was substantial. The NSGA-II algorithm’s lower coverage rate solely analyses the coverage rate in its objective function.

By contrast, the MOGA-AMPazy algorithm incorporates mobility- and sensing-related energy use in addition to the coverage rate in its objective function. MOGA-AMPazy was found to enhance all test coverage rates compared with the SPEA2 algorithm. This improvement of the MOGA-AMPazy approach is based on limiting the average moving distances of all network nodes. From another aspect, the SPEA2 method is based on the penalty function of the most significant moved distance. Furthermore, the MOGA-AMPazy algorithm limits all node sensing radii and saves energy-sensing for all tests. MOGA-AMPazy also ensures population diversity. This algorithm prevents the system from becoming trapped at a local optimum and accelerates its convergence.

For case study 2 (Figure 9), the experiment was conducted to perceive the effect of node density on the dense network. This study observed the impact of node density on the proposed algorithm’s performance compared with the NSGA-II algorithm. The experiment varied the quantity of randomly deployed nodes between 100 and 200 within a \(1000 \times 1000 \times 1000\) field.

Figure 10 depicts the coverage area ratio for the three algorithms versus the number of deployed nodes. The suggested algorithm provides higher and more
stable coverage rates owing to its ability to manage the trade-off between coverage and energy usage. For varying node densities, it consumes less energy than the NSGA-II and MOPSO algorithms. In a densely distributed environment, the NSGA-II algorithm performs well; however, its efficiency diminishes as node density grows.

From another aspect, the MOGA-AMPazy algorithm can modify the sensing radius of all sensor nodes to maximize coverage while conserving energy at varied node densities. The investigation was carried out for 30 cycles to assess algorithm performance. The analysis demonstrated that MOGA-AMPazy achieved a higher percentage accuracy compared with the other algorithms.

**Performance analysis of energy consumption**

Figure 11 compares the energy consumption of sensor nodes of different UWSN algorithms for other numbers of nodes. The MOGA-AMPazy algorithm conserved more energy for the UWSN for every round of the experiment, demonstrating its superiority over the existing algorithms. NSGA-II overcomes the computational difficulty of sorting non-dominated solutions, the absence of elitism and the need to provide a sharing value. Figure 12 compares the energy usage of the four algorithms. The y-axis depicts the energy consumption ratio, and the x-axis shows the four techniques for 30 different data traces. The results show that MOGA-AMPazy consumes 40.93% less energy than the other three algorithms. For all processed data, the results were computed at data traces 1 to 30.

**Performance analysis on Pareto optimal metrics**

This study used five common two-objective ZDT problems – ZDT-1, ZDT-2, ZDT-3, ZDT-4 and ZDT-6 – as benchmark tests\(^\text{29}\) to exploit specific problem characteristics and benchmark the proposed solution and three existing systems. The test instances consisted of seven iterations. The experiment generated 50 approximation fronts for each iteration that competed for survival in the candidate pool. The performance assessment examined IGD, hypervolume and diversity. Each process involved binary tournaments to identify losers to prevent their further consideration. Pareto fronts were reserved in a loser bracket for the next iteration. When all iterations were complete, the benchmark function identified the winner.

The Pareto front obtained by the proposed solution shows almost complete coverage concerning the actual Pareto front. Figure 13 shows the results for the best obtained Pareto optimal values of the proposed MOGA-AMPazy algorithm. This study observed that the developed MOGA-AMPazy performs well with higher accuracy and better robustness. This research requires a decision-maker to choose the optimal compromise option that controls the deployed nodes’ placement based on the proposed solution’s Pareto. Overall, decision-makers are applied to find the best compromise solution using problem-specific information. In the case of the two objectives of sensor node deployment – high coverage and low energy consumption – the selection chose the lowest energy solution below a certain coverage level, which, in this study, was 95%.

**Performance analysis of execution time**

This section presents the performance analysis used to validate the proposed MOGA-AMPazy solution’s execution time against four existing algorithms. The entire execution time of each algorithm was calculated along with the unconstrained test problem. The average execution time for the suggested performance metrics was 2.169 s for a two-objective unconstrained test problem, ZDT 1, if each approximation front has 100 answers and the actual Pareto front has 15 solutions. The maximum and minimum time required was 3.811 and 1.247 s, respectively. Figure 14 shows the execution
This study discovered that different test problems with the same objectives, that is, ZDT 1–ZDT 6, required almost the same execution time.

Figure 13. Obtained Pareto optimal solutions by MOGA-AMPazy for ZDT-1, ZDT-2, ZDT-3, ZDT-4 and ZDT-6.

Figure 14. Execution time of MOGA-AMPazy for each test problem.

Figure 15. Execution time for proposed MOGA-AMPazy compared with three existing algorithms.

Figure 15 illustrates the impact of execution time on the algorithms, namely, that the algorithm proposed in this study achieved a lower execution time than the existing algorithms.
Conclusion and future works

This study proposed a node deployment solution called MOGA-AMPazy to maximize the coverage rates and minimize the energy consumption of mobile UWSNs. This study empirically developed a multi-objective technique solution to solve the multiple and conflicting objectives in the UWSN environment. The method adapted the original NSGA-II by introducing a hybridization of an AMP GA, fuzzy dominance-based decomposition and prospect theory algorithms. The proposed solution measured the coverage rates using two case studies to analyse the effect of the number of sensor nodes and coverage rates. The analysis established that the coverage rates are higher in dense networks compared with sparse networks.

Furthermore, this study measured the overall execution time of the algorithms calculated along with the unconstrained test problem. The results highlight that the algorithm proposed in this study was executed in less time than the existing algorithms. Overall, the results and comparative analysis between the proposed and current multi-objective algorithms prove that the MOGA-AMPazy algorithm solves the multi-objective sensor nodes deployment problem more effectively than the NSGA-II, SPEA2 and MOEA/D algorithms. This study makes several significant contributions to the body of knowledge on UWSNs by providing a functional multi-objective representation of the decision-maker and mission planners in effectively monitoring the region of interest. This research has some limitations – most notably the unavailability of the most recent data. We used openly available validated data sets, but it is worth mentioning that this research would have been more precise had it employed the most recent deployment data sets, such as military and surveillance monitoring data sets. However, these data sets are not available due to privacy and data protection policies. Future works should focus on implementing the proposed solution in real-time multi-objective mobile sensor nodes deployment for the environment, infrastructures and applications of the Internet of underwater things.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship and/or publication of this article.

ORCID iD

Salmah Fattah https://orcid.org/0000-0003-4377-9256

References

1. Vilela J, Kashino Z, Ly R, et al. A dynamic approach to sensor network deployment for mobile-target detection in unstructured, expanding search areas. IEEE Sens J 2016; 16: 4405–4417.
2. Ullah I, Liu Y, Su X, et al. Efficient and accurate target localization in underwater environment. IEEE Access 2019; 7: 101415–101426.
3. Wang H, Li Y, Chang T, et al. Event-driven sensor deployment in an underwater environment using a distributed hybrid fish swarm optimization algorithm. Appl Sci 2018; 8: 1638.
4. Shen W, Zhang C, Zhang X, et al. A fully distributed deployment algorithm for underwater strong k-barrier coverage using mobile sensors. Int J Distrib Sens N 2019; 15: 1550147719838196.
5. Dang X, Shao C and Hao Z. Target detection coverage algorithm based on 3D-Voronoi partition for three-dimensional wireless sensor networks. Mob Inf Syst 2019; 2019: 7542324.
6. Pedlosky J. Ocean circulation theory. Berlin: Springer Science & Business Media, 2013.
7. Mahboubi H and Aghdam A. Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: relocation by virtual force. IEEE Trans Contr Netw Syst 2016; 4: 736–748.
8. Fang W, Song X, Wu X, et al. Novel efficient deployment schemes for sensor coverage in mobile wireless sensor networks. Inform Fusion 2018; 41: 25–36.
9. Liu C, Zhao Z, Qu W, et al. A distributed node deployment algorithm for underwater wireless sensor networks based on virtual forces. J Syst Architect 2019; 97: 9–19.
10. Yakıcı E and Karatas M. Solving a multi-objective heterogeneous sensor network location problem with genetic algorithm. Comput Netw 2021; 192: 108041.
11. Zou J, Gundy S, Kusyk J, et al. Bio-inspired topology control mechanism for autonomous underwater vehicles used in maritime surveillance. In: 2013 IEEE international conference on technologies for homeland security (HST), Waltham, MA, 12–14 November 2013, pp. 201–206. New York: IEEE.
12. Harizan S and Kuila P. Evolutionary algorithms for coverage and connectivity problems in wireless sensor networks: a study. In: Das S, Samanta S, Dey N, et al. (eds) Design frameworks for wireless networks. Singapore: Springer, 2020, pp. 257–280.
13. Singh A, Sharma S and Singh J. Nature-inspired algorithms for wireless sensor networks: a comprehensive survey. Comput Sci Rev 2021; 39: 100342.
14. Du Y, Gong J, Wang Z, et al. A distributed energy-balanced topology control algorithm based on a noncooperative game for wireless sensor networks. Sensors 2018; 18: 4454.
15. Lin C, Deng D and Wang S. Extending the lifetime of dynamic underwater acoustic sensor networks using multi-population harmony search algorithm. IEEE Sens J 2013; 16: 4034–4042.
16. Wang H, Li Y, Chang T, et al. Event-driven sensor deployment in an underwater environment using a distributed hybrid fish swarm optimization algorithm. Appl Sci 2018; 8(9): 1638.
17. Zhang Y, Wang M, Liang J, et al. Coverage enhancing of 3D underwater sensor networks based on improved fruit fly optimization algorithm. Soft Comput 2017; 21: 6019–6029.
18. Lv X, Li H and Li H. A node coverage algorithm for a wireless- sensor-network-based water resources monitoring system. Cluster Comput 2017; 20: 3061–3070.
19. Raj Priyadarshini R and Sivakumar N. Enhancing coverage and connectivity using energy prediction method in underwater acoustic WSN. J Amb Intel Hum Comp 2020; 11: 2751–2760.
20. Wang SC, Hsiao HC, Lin CC, et al. Multi-objective wireless sensor network deployment problem with cooperative distance-based sensing coverage. Mobile Netw Appl 2022; 27(1): 3–14.
21. Jebi RC and Baulkani S. Mitigation of coverage and connectivity issues in wireless sensor network by multi-objective randomized grasshopper optimization based selective activation scheme. Sustain Comput Infor 2022; 35: 100728.
22. Qu Y and Georgakopoulos S. A centralized algorithm for prolonging the lifetime of wireless sensor networks using particle swarm optimization. In: WAMICON 2012 IEEE wireless microwave technology conference, Cocoa Beach, FL, 15–17 April, pp.1–6, 2012. New York: IEEE.
23. Reshko G, Mason M and Nourbakhsh I. Rapid prototyping of small robots. Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University, 2002.
24. Mei Y, Lu Y, Hu Y, et al. Energy-efficient motion planning for mobile robots. In: IEEE international conference on robotics and automation, 2004. Proceedings. ICRA’04.2004, New Orleans, LA, 26 April–1 May 2004, pp.4344–4349. New York: IEEE.
25. Cardei M, Wu J, Lu M, et al. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In: WiMob’2005), IEEE international conference on wireless and mobile computing, networking and communications, 2005, Montreal, QC, Canada, 22–24 August 2005, pp.438–445. New York: IEEE.
26. Dhawan A, Vu C, Zelikovsky A, et al. Maximum lifetime of sensor networks with adjustable sensing range. In: Seventh ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD’06), Las Vegas, NV, 19–20 June 2006, pp.285–289. New York: IEEE.
27. Tversky A and Kahneman D. Advances in prospect theory: cumulative representation of uncertainty. J Risk Uncertainty 1992; 5: 297–323.
28. Barolli L, Koyama A and Shiratori N. A QoS routing method for ad-hoc networks based on genetic algorithm. In: 14th international workshop on database and expert systems applications, 2003. Proceedings. Prague, 1–5 September 2003, pp.175–179. New York: IEEE.
29. Hansen M and Jaszkiewicz A. Evaluating the quality of approximations to the non-dominated set. IMM Technical Report IMMREPI998-7. Lyngby: Technical University of Denmark, 1998.