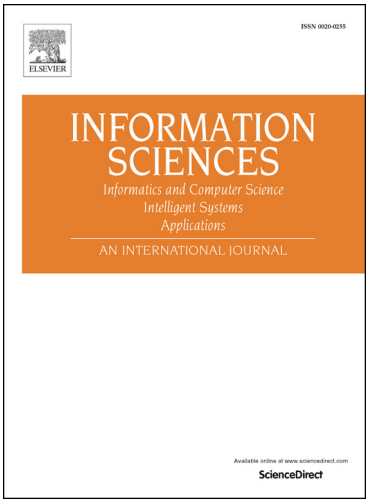
Journal Pre-proof



An efficient particle swarm optimization with evolutionary multitasking for stochastic area coverage of heterogeneous sensors

Shuxin Ding, Tao Zhang, Chen Chen, Yisheng Lv, Bin Xin et al.

PII: S0020-0255(23)00904-0
DOI: https://doi.org/10.1016/j.ins.2023.119319
Reference: INS 119319

To appear in: *Information Sciences*

Received date: 17 October 2022
Revised date: 27 May 2023
Accepted date: 10 June 2023

Please cite this article as: S. Ding, T. Zhang, C. Chen et al., An efficient particle swarm optimization with evolutionary multitasking for stochastic area coverage of heterogeneous sensors, *Information Sciences*, 119319, doi: https://doi.org/10.1016/j.ins.2023.119319.

An efficient particle swarm optimization with evolutionary multitasking for stochastic area coverage of heterogeneous sensors

Shuxin Ding[a,b], Tao Zhang[a,c,*], Chen Chen[d,*], Yisheng Lv[e], Bin Xin[d], Zhiming Yuan[a,c], Rongsheng Wang[f], Panos M. Pardalos[g]

[a]*Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China*
[b]*The Center of National Railway Intelligent Transportation System Engineering and Technology, Beijing 100081, China*
[c]*Traffic Management Laboratory for High-Speed Railway, National Engineering Research Center of System Technology for High-Speed Railway and Urban Rail Transit, Beijing 100081, China*
[d]*National Key Laboratory of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China*
[e]*State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[f]*Scientific and Technological Information Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China*
[g]*Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA*

Abstract

This paper investigates the stochastic area coverage problem of sensors with uncertain detection probability. The risk associated with uncertain parameters is managed using the conditional value-at-risk (CVaR) risk measure. The loss function is represented by the uncovered area coverage rate. We then formulate the minimum CVaR-based uncovered area coverage (CVaR-UAC) problem and provide some theoretical guarantees for the problem. Unlike previous research that treats area coverage as a single problem, we propose an efficient particle swarm optimization (PSO) with evolutionary multitasking to solve the stochastic area coverage problem along with with multiple simplified problem forms. These simplified problems act as the auxiliary tasks for the original CVaR-UAC to enhance the evolutionary search. We have improved the proposed PSO algorithm from the framework of disturbance PSO and virtual force directed co-evolutionary particle swarm optimization, using a hybrid method in population initialization and an adaptive perturbation in individual updating. As a result, the exploration ability of the algorithm is significantly enhanced. The experiment results have demonstrated the effectiveness of the proposed algorithm compared with state-of-the-art algorithms in terms of solution quality.

Keywords: Wireless sensor networks, Stochastic area coverage, Conditional value-at-risk, Co-evolutionary particle swarm optimization, Adaptive perturbation, Evolutionary multitasking

1. Introduction

Wireless sensor networks (WSNs) consist of small, inexpensive, low-powered homogeneous or heterogeneous sensors. In recent years, WSNs have gained significant attention as a research field for various applications, including fire monitoring, battlefield surveillance, target tracking, intrusion detection, and transportation [5, 13]. The primary goal of WSNs is to sense the environment and enable communication among sensors.

The sensor deployment problem is a classical optimization problem that falls under the category of resource allocation problem. Its main objective is to determine the optimal location for sensor nodes. This problem is related to the facility location problem, which focuses on locating the facilities to provide better services to customers (demand

*Corresponding authors.
Email addresses: dingshuxin@rails.cn (Shuxin Ding), 13701193534@139.com (Tao Zhang), xiaofan@bit.edu.cn (Chen Chen), yisheng.lv@ia.ac.cn (Yisheng Lv), brucebin@bit.edu.cn (Bin Xin), zhimingyuan@hotmail.com (Zhiming Yuan), rshwang@outlook.com (Rongsheng Wang), pardalos@ufl.edu (Panos M. Pardalos)

points) with better coverage or fewer facilities. Sensors are considered facilities, while targets are demand points. The covering problem is a popular facility location model [12]. The distance between customers and facilities determines the services customers receive from facilities.

Coverage problems can be classified into three models: point/target coverage, area coverage, and barrier coverage [5]. Point coverage deals with a set of target points, where each point should be observed by sensors. In area coverage, the entire area is considered. Barrier coverage deals with intruders who attempt to penetrate an area. This paper focuses on the area coverage problem, with the main objective of determining the optimal location of the sensor network to monitor an area.

The sensor detection model describes the sensing ability and quality of the geometric relation between a target point and a sensor node. In WSNs, there are two sensor detection models: the binary detection model and the probabilistic detection model [49]. In the binary detection model, a target is deterministically detected when it is within the sensing range. However, the detection probability of the sensor may be uncertain due to some noise and obstacles. In the probabilistic sensor detection model, the detection probability decays with distance from the sensor. In our work, we use the probabilistic sensor detection model.

The deployment space of a sensor deployment problem can be classified into two types: continuous space-based deployment and discrete space-based (or grid-based) deployment [8]. In continuous space-based deployment, nodes can be placed anywhere within the deploy region. In discrete space-based deployment, nodes can only be placed in the grid network or predefined candidate positions. This paper investigates continuous space-based deployment.

Regardless of the type of sensor deployment, the coverage problem in WSNs is nondeterministic polynomial complete (NP-complete) [13]. Metaheuristics are commonly used to solve this problem [8]. Among these algorithms, particle swarm optimization (PSO) is most frequently used [20, 21]. Parallel particle swarm optimization produces the same results with multiple processors, reducing the computation time [3]. In [42], a parallel particle swarm optimization divides the sensing area and the sensors equally into several parts to deal withhomogeneous sensors to be deployed [42]. The partitioned search space decreases the complexity of the problem. Ding *et al.* [7] proposed a disturbance PSO (d-PSO) with a Gaussian perturbation to update the velocity, which shows fast convergence. Ning *et al.* [30] described a discrete and multi-swarm PSO to solve the dynamic deployment of WSN, maximizing the coverage while minimizing the moving distance. Tang *et al.* [39] established a three-dimensional heterogeneous sensor network model and provided an improved PSO algorithm. Wang *et al.* [44] proposed a resampled PSO (RPSO) to maximize the coverage and energy efficiency of the WSN.

Some approaches combine other methods with metaheuristics to find good solutions. In [1], the Voronoi diagram was combined with PSO to obtain the best coverage. This algorithm used PSO to find optimal positions, while the Voronoi diagram evaluated the solution's fitness value. Some studies used virtual force between sensors, environment, and obstacles to improve the sensor coverage rate [43]. This algorithm was called Virtual Force Algorithm (VFA). Liang *et al.* [26] used a virtual force based coverage algorithm to achieve area coverage. Directional sensors were subjected to four forces caused by neighbor sensors and uncovered regions. Wang *et al.* [43] presented an improved co-evolutionary PSO algorithm (VFCPSO) that combines virtual force and PSO with a co-evolutionary mechanism to solve the dynamic sensor deployment problem. Yoon and Kim [46] proposed an efficient genetic algorithm (GA) to maximize the coverage deployment of heterogeneous sensors. A normalization method was proposed by analyzing that the phenotype space of the problem is a quotient space of the genotype space. Besides, a local search for GA is applied by VFA. Yoon and Kim [47] recently improved their former studies [46] by providing a new coverage estimation method and an iterative local search to develop a memetic algorithm (MA). Hanh *et al.* [18] proposed an improved GA with heuristic population initialization and exact integral area calculation for fitness function. Nguyen *et al.* [29] proposed a novel coverage strategy for WSN by Ions Motion Optimization (IMO) algorithm.

Most studies reviewed above consider deterministic sensor deployment problems with parameters having exact values. However, many parameters are uncertain in the real environment, such as the sensor's detection probability and target locations [8, 11]. Erişkin [11] proposed a target coverage problem with uncertainties in the target locations. GA was developed to provide a robust solution to the problem. This paper considers uncertainties in the sensor's detection probability, where the sensor type considered is probabilistic rather than binary. The scenario approach can tackle the uncertainty of sensor detection. Furthermore, based on current research, when the number of sensors increases to a large scale, most metaheuristics quickly converge to a local minimum, making it challenging to obtain optimal global solutions. Table 1 summarizes the reviewed literature on the sensor deployment problem.

Evolutionary multitasking (EMT) has recently provided a new paradigm for evolutionary computation, which

Table 1: Summary of the literature on sensor deployment problem.

Ref.	Cov. Type	Sensor Type	Detection Model	Deployment Space	Optimization Objective	Solution Method
[7]	Area	Homogenous	Probabilistic	Continuous	Coverage	d-PSO
[42]	Area	Heterogeneous	Binary	Continuous	Coverage	Parallel PSO
[30]	Area	Heterogeneous	Probabilistic	Continuous	Coverage and energy consumed	Multi-swarm PSO
[39]	Area & Target	Heterogeneous	Probabilistic	Continuous	Coverage and target detection probability	Improved PSO
[44]	Area	Heterogeneous	Binary	Continuous	Coverage	RPSO
[26]	Area	Homogenous	Binary	Continuous	Coverage and energy consumed	VFA
[43]	Area	Heterogeneous	Probabilistic	Continuous	Coverage	VFCPSO
[46]	Area	Heterogeneous	Binary	Continuous	Coverage	GA
[47]	Area	Heterogeneous	Binary	Continuous	Coverage	MA
[18]	Area	Heterogeneous	Binary	Continuous	Coverage	GA
[38]	Target	Heterogeneous	Probabilistic	Discrete	Coverage	LINGO
[49]	Area	Homogenous	Probabilistic	Continuous	Coverage	VFA
[24]	Area & Target	Homogenous	Probabilistic	Continuous	Coverage	VFA
[29]	Area	Homogenous	Probabilistic	Continuous	Coverage	IMO
[34]	Area	Heterogeneous	Binary	Continuous	Coverage and connectivity	GA with steepest descend
[4]	Area	Homogenous	Binary	Continuous	Coverage	Levy Flight mechanism with WOA
[2]	Area	Homogenous	Binary	Continuous	Coverage	GSO with Voronoi and K-means
[11]†	Target	Heterogeneous	Probabilistic	Discrete	Coverage	GA
This paper†	Area	Heterogeneous	Probabilistic	Continuous	Coverage	Improved PSO with EMT

WOA: whale optimization algorithm; GSO: glowworm swarm optimization.
† References with uncertainty characteristics. [11]: probabilistic target locations. This paper: coverage under uncertainty.

deals with multiple optimization problems simultaneously [16, 25]. The knowledge obtained in different tasks is transferred by EMT during the evolution process. As a result, the searching ability and convergence performance of the algorithm on each task are greatly enhanced [31]. EMT has been successfully applied in various multitasking optimization problems, such as unmanned systems planning [17], WSN lifetime maximization [37], multiple robot navigation [23], etc. Besides, EMT shows promising prospects in solving the large-scale optimization problem by constructing simplified versions of the original problem as the auxiliary or helper tasks [14]. By generating a multitasking environment, the knowledge from different auxiliary tasks enhances the global search in the main task [31, 33]. Therefore, for the sensor deployment problem, different simplified deployment problems can be generated as auxiliary tasks, and multitasking search on these tasks can help enhance the search on the sensor deployment problem. To the best of our knowledge, the sensor deployment problem has not been solved through EMT.

In this paper, we address the problem of determining the deployment scheme of a set of heterogeneous sensors given a target area to cover. Since the sensor's detection probability is uncertain, we use the conditional value-at-risk (CVaR) measure to control the uncovered area coverage rate of different scenarios [32]. CVaR is a percentile risk measure that takes the average of the loss over part of the worst-case values. This risk measure avoids the over-conservativeness of the solution. It has been applied to several sensor networks-related works [35], e.g., sensor scheduling and communication network interdiction. However, this risk measure has not previously been used in sensor coverage problems to the best of our knowledge. Therefore, we formulate this sensor deployment problem as the minimum CVaR-based uncovered area coverage (CVaR-UAC) problem. We summarize our contributions as follows:

- We present a mathematical model for the minimum CVaR-UAC problem to deal with the area coverage for heterogeneous sensor deployment with uncertainty in sensor detection. Different scenarios are generated by the

uncertainty of the sensors' parameters with probabilistic sensors.

- We provide an approximation ratio on the discrete version of CVaR-UAC by the proposed PSO. The submodularity and monotonicity of the coverage function help to obtain the results.
- The CVaR-UAC is solved by an efficient PSO search based on the evolutionary multitasking technique. Multiple simplified CVaR-UAC$_S$ are generated by sensor grouping and area partitioning. The efficient PSO search is conducted simultaneously on the original CVaR-UAC and the generated multiple simplified ones. This approach significantly improves the deployment performance with the help of searching in simplified problems.
- We propose an efficient PSO algorithm for CVaR-UAC with the following two improvements:
  - We propose a hybrid initialization strategy incorporating the VFA with the random initialization strategy. Numerical results show that the proposed algorithm with hybrid initialized populations can find better solutions than randomly initialized ones. It provides better initial solutions, making it easier to achieve global convergence.
  - We add adaptive perturbation to the d-PSO and VFCPSO framework, thus creating the proposed algorithm, ad-VFCPSO. The exploration of the proposed algorithm is highly improved, and the experiments show that ad-VFCPSO outperforms some state-of-the-art algorithms with respect to the optimization goal.

The rest of this paper is organized as follows. Section 2 describes the minimum CVaR-UAC problem. Section 3 presents our algorithm based on EMT and adaptive perturbation with the VFCPSO framework. Experimentation design with algorithms and experiment settings are given in Section 4. The performance of the proposed algorithm is evaluated in Section 5. Finally, conclusions and future work are given in Section 6.

2. Preliminaries and problem formulation

This paper analyzes the area coverage problem based on probabilistic sensors under uncertainty controlled by CVaR, which involves deploying sensors with risk preference for area coverage. Firstly, we describe the detection model and area deployment scenarios based on the probabilistic sensors. Then, we define the minimum CVaR-UAC problem based on the probabilistic detection model. Risk management techniques are adopted to deal with the stochastic scenarios of area coverage. We aim to minimize the CVaR of the uncovered area coverage rate with the given probabilistic sensors.

2.1. Detection model

As mentioned before, we use a probabilistic sensor detection model. The probability of a grid point (x, y) being covered by a sensor node s_i can be computed as [24].

$$c_{xy}(s_i) = \begin{cases} 0 & \text{if } r + r_e \le d(s_i, P) \\ e^{(-\alpha_1 \lambda_1^{\beta_1} / \lambda_2^{\beta_2} + \alpha_2)} & \text{if } r - r_e < d(s_i, P) < r + r_e \\ 1 & \text{if } d(s_i, P) \le r - r_e \end{cases} \tag{1}$$

where $d(s_i, P)$ denotes the Euclidean distance between sensor node s_i and point P, the constant r is the sensing range, $r_e (r_e < r)$ measure the uncertainty of the detection. λ_1 and λ_2 are the input parameters, which measure the difference of distance $d(s_i, P)$ with the boundary of the judging condition of Eq. (1) [29]. They are calculated by $\lambda_1 = r_e - r + d(s_i, P)$ and $\lambda_2 = r_e + r - d(s_i, P)$. α_1, α_2, β_1 and β_2 are detection probabilities parameters. These values vary depending on the sensor's types and characteristics.

2.2. Problem statement

There are N sensors deployed in an 2D plane of size $L \times L$. The target area is divided into $\frac{L}{grid} \times \frac{L}{grid}$ grids, where each grid has an area of $grid^2$. Each grid can be monitored by multiple sensors nearby in a collaborative manner. Let S denote the set of sensor nodes, sensor node s_i be the sensor located at point (x_i, y_i), and point P represent any grid point (x, y). Each grid point covered by a sensor will have a coverage rate, and the grid covered by multiple sensors may have a better coverage rate. The probability that $P(x, y)$ is covered by a set of sensors S is defined as:

$$c_{xy}(S) = 1 - \prod_{i=1}^{N}(1 - c_{xy}(s_i)) \tag{2}$$

This paper employs CVaR to deal with uncertainty in the area coverage problem. The loss function used is the uncovered rate of the sensors. Therefore, we define the minimum CVaR-UAC problem: given a set of sensors S and a target area with $L \times L$ grids, the objective is to minimize the CVaR of the uncovered rate in the area. Our aim is to determine the position of different sensors in the area. The uncertainty of the detection r_e is considered an uncertain value with a known distribution, and a scenario-based approach is adopted to deal with uncertainty.

In the next step, we define the uncovered area coverage rate and CVaR of the uncovered area coverage rate, which are used as metrics for the performance evaluation of the WSN deployment.

Definition 1. *(Uncovered area coverage rate). The uncovered area coverage rate (UACR) of a sensor node set S is defined as:*

$$UACR(S) = 1 - \frac{Area(S)}{L * L} = 1 - \frac{\sum_{x=0}^{L}\sum_{y=0}^{L} c_{xy}(S)}{L * L} \tag{3}$$

where $Area(S)$ is the coverage area of the sensor node set S, $L * L$ is the total area of the target region. A suitable sensor deployment scheme should have a lower UACR. Thus, the larger $Area(S)$ is, the better quality of the sensor deployment. Therefore, UACR can be used as a loss function when risk measure is considered.

CVaR [32] is a risk measure closely related to a well-known quantitative risk measure referred to as Value-at-Risk (VaR). For a given loss distribution, β-VaR denotes the smallest value α such that the loss does not exceed α with probability $1 - \beta$, while β-CVaR represents the expected loss conditional on the loss exceeding β-VaR.

To better describe the relationship between β-VaR and β-CVaR, their representations are shown in Fig. 1. The figure displays the frequency of a loss distribution with a long tail, showing the mean value, VaR, CVaR, and maximum loss are shown in the figure, with $\beta = 0.95$. Since β-CVaR is the expected loss conditional on the loss exceeding β-VaR, it represents the average loss value between VaR and maximum loss. Using only VaR may ignore those bad scenarios near the maximum loss.

Remark 1. *When $\beta = 1$, the decision maker is the most risk-averse. We can see in Fig. 1 that both VaR and CVaR will equal the maximum loss. When $\beta = 0$, the decision maker is risk-neutral. VaR will equal the minimum loss and CVaR will equal the mean value.*

Therefore, CVaR is a more conservative risk measure than VaR. With the loss function UACR, we have the following definition:

Definition 2. *(Conditional value-at-risk of uncovered area coverage rate). The CVaR of the UACR is an effective metric for evaluating deployment performance when considering uncertainty and risk. A finite set of scenarios $\xi \in \Xi$ with a total number W is used to formulate different realizations of the sensor network parameters. The CVaR of the UACR of a sensor node set S is defined as:*

$$CVaR_{\beta}(UACR(S, \xi)) = \min_{\alpha \in \mathbb{R}} \left\{ \alpha + \frac{1}{1-\beta} \sum_{w=1}^{W} p_w \left[UACR(S, \xi_w) - \alpha \right]^+ \right\} \tag{4}$$

where ξ_w denotes the uncertain parameters of the WSNs in the wth scenario, p_w is the probability of scenario ξ_w, and $[t]^+ = \max\{t, 0\}$. β is the probability level that controls the risk preference or the level of conservatism. When β equals 1, Eq. (4) corresponds to minimizing the maximum of the UACR, while β equals 0, Eq. (4) corresponds to minimizing the average value of the UACR. Similarly, A good sensor deployment scheme should have a lower CVaR of UACR.

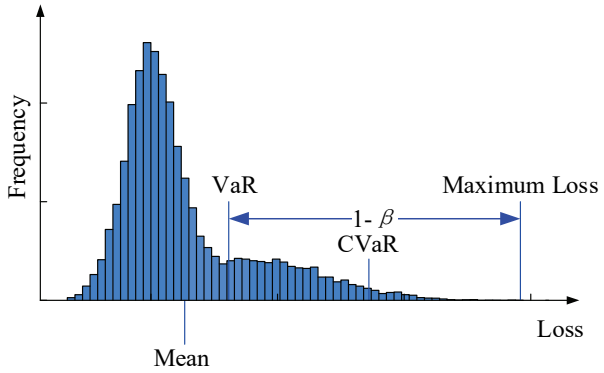


Fig. 1: The frequency of a loss distribution to show mean value, VaR, CVaR, and maximum loss, $\beta = 0.95$.

Therefore, we proposed the minimum CVaR-UAC problem:

$$\min CVaR_{\beta}(UACR(S,\xi)) \tag{5}$$

We consider the value p_w can all be set to $1/W$, so each scenario has the same probability. This problem can be easily transformed for easy implementation by introducing extra variables u_w and constraints.

$$\min_{\alpha} \left\{ \alpha + \frac{1}{W(1-\beta)} \sum_{w=1}^{W} u_w \right\} \tag{6}$$

$$\text{s.t.} \quad u_w \geq UACR(S,\xi_w) - \alpha \qquad \forall w = 1,\ldots,W \tag{7}$$

$$u_w \geq 0 \qquad \forall w = 1,\ldots,W \tag{8}$$

$$\alpha \in \mathbb{R} \tag{9}$$

It is easy to conclude that the CVaR-UAC is NP-hard [5].

2.3. Theoretical guarantees

We first consider a particular situation for CVaR-UAC, where the sensor coverage is limited to finite locations and the optimization objective is the maximum, which is a discrete version of the maximum CVaR-based area coverage (D-CVaR-AC) problem. The presence of finite locations for sensor coverage means that there is a set of finite candidate locations for deploying sensors. The sensor coverage problem is usually considered as maximizing the covered area with a limited number of selected sensors. This problem belongs to submodular set optimization, which plays an important role in combinatorial optimization [40]. We will take advantage of the properties from submodular set functions to provide the submodularity and theoretical guarantees of D-CVaR-UAC.

Lemma 1. *The maximum D-CVaR-AC problem is monotone, submodular.*

Proof. We utilize Eq. (2) to prove the lemma. Eq. (2) describes a basic coverage function for sensor coverage. Typically, a sensor coverage problem is maximizing the coverage, where the coverage function is a submodular set-functions [36]. In addition to submodularity, the coverage function is also monotone in the number of sensors [40].

According to the definition of CVaR, it measures the expected value with a confidence level within $1-\beta$ for W different scenarios. Thus, the CVaR of area coverage rate (ACR) can be obtained by averaging the ACR on a subset of all scenarios, which is a linear combination of ACR for a set of sensor S. The ACR can be calculated as $ACR(S) = 1 - UACR(S) = \frac{\sum_{x=0}^{L}\sum_{y=0}^{L} c_{xy}(S)}{L*L}$, which is a positive linear combinations of coverage function. Therefore, as suggested in [28], a positive linear combination of submodular functions is submodular, and the D-CVaR-AC problem is also submodular.

Regarding monotonicity, it is easy to observe that the D-CVaR-AC is monotone as a positive linear combination of coverage functions. □

Theorem 1. *For the maximum D-CVaR-AC, Algorithm 2 achieves an approximation ratio of* $1 - e^{-1}$.

Proof. We follow the proof of [28] and [15] to analyze the approximation ratio of the proposed algorithm. In [28], it is shown that a suboptimal solution by the greedy algorithm can achieve an approximation ratio of $1 - e^{-1}$ for monotone submodular optimization problems. Moreover, in [15], a global simple evolutionary multiobjective optimizer (GSEMO) can achieve a $\left(1 - e^{-1}\right)$-approximation based on [28]. GSEMO is an evolutionary optimization framework with local search operations and greedy behavior, which has the same approximation ratio as the greedy algorithm. The proposed Algorithm 2 also exhibits excellent search abilities due to its co-evolutionary, adaptive perturbation, and EMT techniques. These techniques divide the decision space, jump out of local optima, and decompose the problem into subproblems, providing global search and maintain diversity during the evolution. Therefore, the proposed algorithm is less likely to end up in poor local optima compared to traditional PSO and (1+1) EA, which only have an approximation ratio of 1/2 (in (1+1) EA [15]). □

Proposition 1. *For the discrete version of the minimum CVaR-UAC (D-CVaR-UAC) problem, Algorithm 2 achieves an approximation ratio of* $1 + \frac{1-f(opt)}{f(opt)}e^{-1}$*, where* $f(opt)$ *is a value of feasible optimal solution.*

Proof. Let $f(S)$ and $g(S)$ represent the function of D-CVaR-UAC and D-CVaR-AC, respectively. We have $f(S) = 1 - g(S)$. According to Theorem 1, we have $g(S) \geq (1 - e^{-1})g(opt)$. This lead to

$$1 - f(s) \geq (1 - e^{-1})(1 - f(opt)).$$

Therefore, the solution obtained by Algorithm 2 with a value

$$\begin{aligned} f(s) &\leq 1 - (1 - e^{-1})(1 - f(opt)) = 1 - (1 - e^{-1} - f(opt) + e^{-1}f(opt)) \\ &= e^{-1} + f(opt) - e^{-1}f(opt) = \frac{e^{-1}}{f(opt)}f(opt) + (1 - e^{-1})f(opt) \\ &= \left(1 + \frac{1 - f(opt)}{f(opt)}e^{-1}\right)f(opt). \end{aligned}$$

□

3. The proposed method

Since the minimum CVaR-UAC problem is NP-hard, there is no polynomial-time algorithm to obtain the exact solution. Therefore, metaheuristics are applied to solve this problem with an approximate solution. PSO has been widely used to deal with the sensor deployment problem. It is a swarm-based intelligent method inspired by the social behavior of a flock of birds developed by Kennedy and Eberhart [20]. However, some drawbacks of using PSO are as follows:

- Since the particle is updated based on its personal best particle and the global best particle, the algorithm will quickly converge to local minima and cause premature convergence. As a result, the particles' diversity is hard to maintain, especially for those large-scale, high-dimensional problems.
- PSO lacks the ability to escape from local minima when all the particles are converged to the same local optimal solution.

As discussed in [14], for large-scale, high-dimensional problems, existing metaheuristic approaches can be classified into three types: (1) decision variable decomposition using the divide-and-conquer mechanism; (2) developing new evolutionary operators; (3) transformation-based methods by generating and solving new small-scale problems. Therefore, our approach considers all three types: (1) VFCPSO is used as the basic search engine; (2) In order to improve the algorithm, the population is initialized using a hybrid method of random and heuristic initialization. An adaptive disturbance technique is utilized in VFCPSO (ad-VFCPSO) for individual updating; (3) Simplified CVaR-UAC$_S$ problems will be generated as auxiliary or helper tasks of the original problem by sensor grouping and area partitioning, enabling the use of EMT to solve the CVaR-UAC problem.

In this section, we introduce the proposed efficient PSO with EMT for CVaR-UAC in detail, including how to construct the simplified CVaR-UAC$_S$ from the original CVaR-UAC, how to initialize the population, how to update the individuals of the particle swarm by ad-VFCPSO, and how to transfer knowledge from the simplified CVaR-UAC$_S$ to the original CVaR-UAC using EMT-ad-VFCPSO (ad-VFCPSO with evolutionary multitasking).

3.1. Simplified CVaR-UAC$_S$ construction

We construct the simplified CVaR-UAC$_S$ problem by grouping sensors and partioning the area of the original CVaR-UAC problem. The original problem is decomposed into simplified versions. For a CVaR-UAC, there are N sensors to be deployed in WSNs, and the position for each sensor is described using a coordinate system as (x_i, y_i). Therefore, a real-coded encoding scheme is adopted. A solution for N sensors can be represented as $(x_1, y_1, x_2, y_2, x_3, y_3, \ldots, x_N, y_N)$, which is $2N$-dimensional for N sensors. This individual representation is frequently used for coverage problems in WSNs [18], and can be easily utilized in PSO-based algorithms.

Random or heuristic methods are used to group sensors and partition the area to construct the simplified CVaR-UAC$_S$. The number of different groups or areas is defined as N_G, resulting in N_G simplified CVaR-UAC$_S$. Each sensor is subjected to the following constraints:

$$\sum_{k=1}^{N_G} z_{ik} = 1 \tag{10}$$

where $z_{ik} = 1$ if sensor i assigned to area/group k, otherwise, $z_{ik} = 0$. Eq. (10) denotes that each sensor should be assigned to one area/group. When the area is divided into N_G small areas, and the constraints in Eq. (10) are satisfied for all sensors, the construction of the CVaR-UAC$_S$ is finished. The process of the proposed construction of CVaR-UAC$_S$ is illustrated in Fig. 2. Without loss of generality, the value of N_G can be set as needed. We select $N_G = 4$ in this paper. In Fig. 2, there are $N = 23$ sensors with three types, in which $N_1 = 4$ for type I, $N_2 = 6$ for type II, and $N_3 = 13$ for type III with sensing range, r_1, r_2, and r_3, respectively. The number of groups/areas is $N_G = 4$.

With a particular sensor grouping and area partitioning configuration, we only need to determine the position of sensors in each group in the assigned area. The encoding scheme for the individuals is shown in Fig. 3. It should be noted that simplified CVaR-UAC$_S$ act as helper tasks for the original CVaR-UAC. Therefore, the whole individuals are divided into two parts, one for simplified CVaR-UAC$_S$, and the other for CVaR-UAC. As shown in Fig. 3, the assignment of sensors to areas/groups can only be seen in the individuals for simplified CVaR-UAC$_S$. The sensor types and the area/group assignments in Fig. 3 are determined according to Fig. 2.

For the first part of individuals for simplified CVaR-UAC$_S$, each element represents the sensor position (x_i, y_i) and the area/group assignment z_{ik}. For example, Fig. 3 shows the position of sensor 7 and the assignment of sensor 4 to area/group 4. For the second part of individuals for CVaR-UAC, each element only represents the sensor position.

CVaR-UAC$_S$ related to different sensor groups and areas for deployment is identified by the task number k ($1 \leq k \leq N_G$). The basic properties of the simplified CVaR-UAC$_S$ are summarized as follows.

- N_p^{emt}: The population size of individuals for CVaR-UAC$_S$. Since N_p is the total population size, we denote $N_p^{main} = N_p - N_p^{emt}$ as the population size of individuals for CVaR-UAC in EMT.
- N_G: The total number of simplified CVaR-UAC$_S$ tasks.
- $\mathbf{f_k^{emt}} = [f_k^{emt1} \quad f_k^{emt2}]$: The objective functions of CVaR-UAC$_S$ for task k. There are two optimization objectives for a simplified CVaR-UAC$_S$: the CVaR of the UACR in area k, which is f_k^{emt1}, and the CVaR of the UACR in areas other than area k, which is f_k^{emt2}.
- $task(k).D$: The dimension of the k^{th} CVaR-UAC$_S$.
- $task(k).x_{ij}$: The position of the i^{th} particle in dimension j for task k.
- $task(k).p_{ij}$: The best position of the i^{th} particle in dimension j for task k.
- $task(k).p_{gj}$: The best position of the whole particles in dimension j for task k.

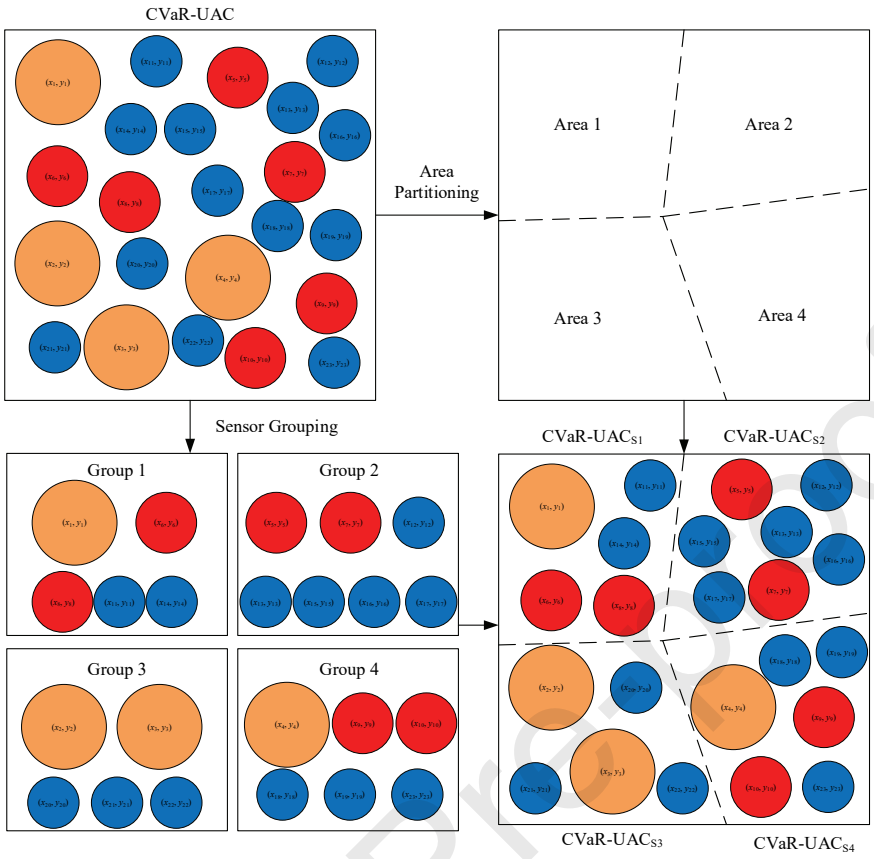


Fig. 2: Constructing simplified CVAR-UAC$_S$ by sensor grouping and area partitioning.

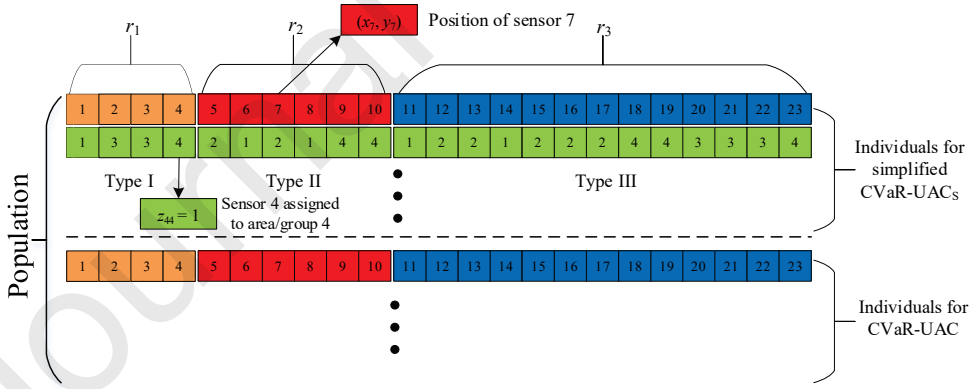


Fig. 3: Encoding scheme of the individuals in the population.

- $task(k).b(j, z)$: The best position of the whole particles with dimension j subsititute by z for task k.

Remark 2. *When comparing two deployment schemes (S_1 and S_2) of CVaR-UAC$_S$ for task k, scheme S_1 is better than scheme S_2 when ($f_k^{emt1}(S_1) > f_k^{emt1}(S_2)$) or ($f_k^{emt1}(S_1) = f_k^{emt1}(S_2)$ and $f_k^{emt2}(S_1) > f_k^{emt2}(S_2)$).*

Algorithm 1 Hybrid Initialization Based on VFA

Input:

r_i: Sensing range of sensor i

w_A, w_R, d_{th}, C: Parameters for VFA

$MaxLoop$: Maximum number of loops for adjusting sensor locations using VFA

f: Objective function

Output:

IND: Individual solution after hybrid initialization

Generate the individual solution IND randomly.
$IND_{new} = IND$.
Set $loops = 1$.
while $loops \leq MaxLoop$ **do**
    Calculate virtual force F using w_A, w_R, d_{th}, C on each sensor.
    $IND_{new} = IND_{new} + F/10$.
    Refine the position of each sensor from individual IND_{new} within the preferred deployment area of the sensor.
    Calculate fitness value of individual $f(IND_{new})$.
    if $f(IND_{new}) \leq f(IND)$ **then**
        $IND = IND_{new}$.
    end if
    $loops = loops + 1$.
end while
return

3.2. Population initialization

In most studies, individuals of the population for evolutionary algorithms (EAs) are usually initialized randomly to ensure diversity. However, these initial solutions are often of low quality, making it difficult for algorithms to converge to optimal or near-optimal solutions. Heuristic initialization for EAs has been applied to many complex optimization problems [45]. An initial population with good quality generally provides better solutions for the optimization problem. However, if the entire population is provided by the heuristic solutions, the population may lack diversity, which reduces the ability to obtain new solutions. Therefore, in our PSO-based algorithm, each individual in the population (both individuals for simplified CVaR-UAC$_S$ and CVaR-UAC) is generated with random and heuristic initialization.

We develop the initialization method by taking inspiration from the popularly used algorithms for WSNs, i.e., VFA [49]. The location of each sensors is adjusted based on the force from other sensors that either push away or pull toward it, depending on the distance. We use random initialization to obtain different initial locations. Then, the locations are adjusted by VFA, and a hybrid initialization method is developed. As a result, these solutions have good quality for coverage and randomness for diversity. Specifically, a solution can be obtained by Algorithm 1. We adopt the VFA in [43], which only considers attractive force between sensors within a certain range. However, the repulsive force in Algorithm 1 differs from [43]. It is proportional to the difference between the sensors' distance d_{ij} and a threshold d_{th} ($|F| = w_R(d_{th} - d_{ij})$, where w_R is a parameter for repulsive force), similar to the attractive force ($|F| = w_A(d_{ij} - d_{th})$, where w_A is a parameter for attractive force). The preferred deployment area for sensor i is restricted based on the individual type. If the individual is used for simplified CVaR-UAC$_S$, the preferred deployment area is the assigned subarea after partitioning. If the individual is used for the original CVaR-UAC, the preferred deployment area is $[0, L] \times [0, L]$ (L is the side length of the deployment area). The maximum number of loops for VFA is set to $MaxLoop = 20$ based on empirical study.

3.3. ad-VFCPSO

d-PSO and VFCPSO are two efficient algorithms used to tackle the sensor deployment problem [7, 43]. The disturbance operator in d-PSO is a Gaussian perturbation that prevents local convergence. However, it may not always be effective during the entire optimization process. Similar to the limitations of using a single mutation operator [22], the best disturbance results cannot be achieved only by one perturbation strategy, but rather by combining

different perturbation strategies at different stages for the best performance. The Gaussian perturbation operator used in the disturbance operator is suitable for local search [41]. Inspired by the adaptive mutation operator proposed in [22, 41], an adaptive disturbance is introduced by adaptively selecting three distributions (i.e., Gaussian, Cauchy, and Lévy). The first perturbation is suited for local search, while the latter two are suited for global search. The adaptive disturbance is added as a perturbation to the velocity item, where it is added to the global best value in [22], and both the global best value and personal best value in [41].

Therefore, we improve d-PSO with an adaptive disturbance and VFCPSO to obtain ad-VFCPSO. Eqs. (11) and (12) are used to update the velocity and position of a particle in ad-VFCPSO.

$$v_{id}^{t+1} = c_0 * n_i^a + c_1 * r_{1i} * (p_{id}^t - x_{id}^t) + c_2 * r_{2i} * (p_{gd}^t - x_{id}^t) + c_3 * r_{3i} * g_{id}^t \tag{11}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{12}$$

where v_{id}^{t+1} and x_{id}^{t+1} are the velocity and position of the i^{th} particle in dimension d at time $t+1$, respectively. p_{id}^t and p_{gd}^t are the best position of the i^{th} particle and the whole particles in dimension d at time t, respectively. c_0 is the disturbance factor that controls the effect of the adaptive disturbance. c_1 and c_2 are acceleration constants, also known as the cognitive factor and social factor. These constants control the motion of the particle to its personal best position and global best position, respectively. c_3 is the acceleration constant that controls the particle by the virtual force. r_1, r_2, and r_3 are random values with uniform distribution [0, 1]. n_i^a is the adaptive disturbance operator of the i^{th} particle selected from three distributions (i.e., Gaussian, Cauchy, and Lévy). The Gaussian distribution used here is $G(\mu = 0, \sigma = 1)$. Cauchy distribution is $C(x_0 = 0, s = 1)$ and Lévy distribution is $L(\mu = 0, c = 1)$ (can be calculated by Mantegna's algorithm [27] as a symmetric Lévy stable distribution). All three distributions will be multiplied by a "disturbance learning rate" δ. Following the elitist learning strategy [48], it is suggested that δ be linearly decreased with the number of the iteration, which is

$$\delta = \delta_{max} - (\delta_{max} - \delta_{min}) \times \frac{t}{MaxGen} \tag{13}$$

where δ_{max} and δ_{min} are the upper bound and lower bound of δ. The values of the bounds are set to $\delta_{max} = 1.0$ and $\delta_{min} = 0.1$ respectively by empirical study. $MaxGen$ is the maximum number of generations.

The proposed adaptive disturbance operator utilizes the three perturbation operators described above. Each operator has a selection ratio denoted by pg, pc, and pl, which satisfies $pg + pc + pl = 1$. Initially, all values are set to 1/3, meaning that the probability of selecting each operator is equal. These values are updated during evolution as follows.

$$pg = \gamma + (1 - 3 \cdot \gamma) \cdot \frac{suc_g}{suc_d} \tag{14}$$

$$pc = \gamma + (1 - 3 \cdot \gamma) \cdot \frac{suc_c}{suc_d} \tag{15}$$

$$pl = \gamma + (1 - 3 \cdot \gamma) \cdot \frac{suc_l}{suc_d} \tag{16}$$

where suc_g is the successful number of Gaussian perturbations, suc_c is the successful number of Cauchy perturbations, suc_l is the successful number of Lévy perturbations, suc_d is the successful number of disturbance operations, and $suc_g + suc_c + suc_l = suc_d$. These values are updated along with particles' personal best value p_i. The parameter γ is the minimum selection ratio set to 0.05 [41].

g_{id}^t is the motion suggested by virtual force of the i^{th} particle in dimension d. For more information on virtual force and VFCPSO, please refer to [43].

In addition to adopting the velocity and position updating Eqs. (11) and (12), ad-VFCPSO utilizes a co-evolutionary manner. The solution vector of a particle is divided into small values, which is the key idea from co-evolutionary particle swarm optimization (CPSO) [6]. Therefore, a $2N$-dimensional vector is split into a 1-D problem for $2N$ swarms. For more details, readers can refer to [6, 7, 43].

3.4. EMT-ad-VFCPSO

Fig. 4 illustrates the framework of EMT-ad-VFCPSO. The problem of CVaR-UAC is first decomposed by sensor grouping and area partitioning to obtain N_G simplified CVaR-UAC$_S$. Then, the individuals of the simplified CVaR-

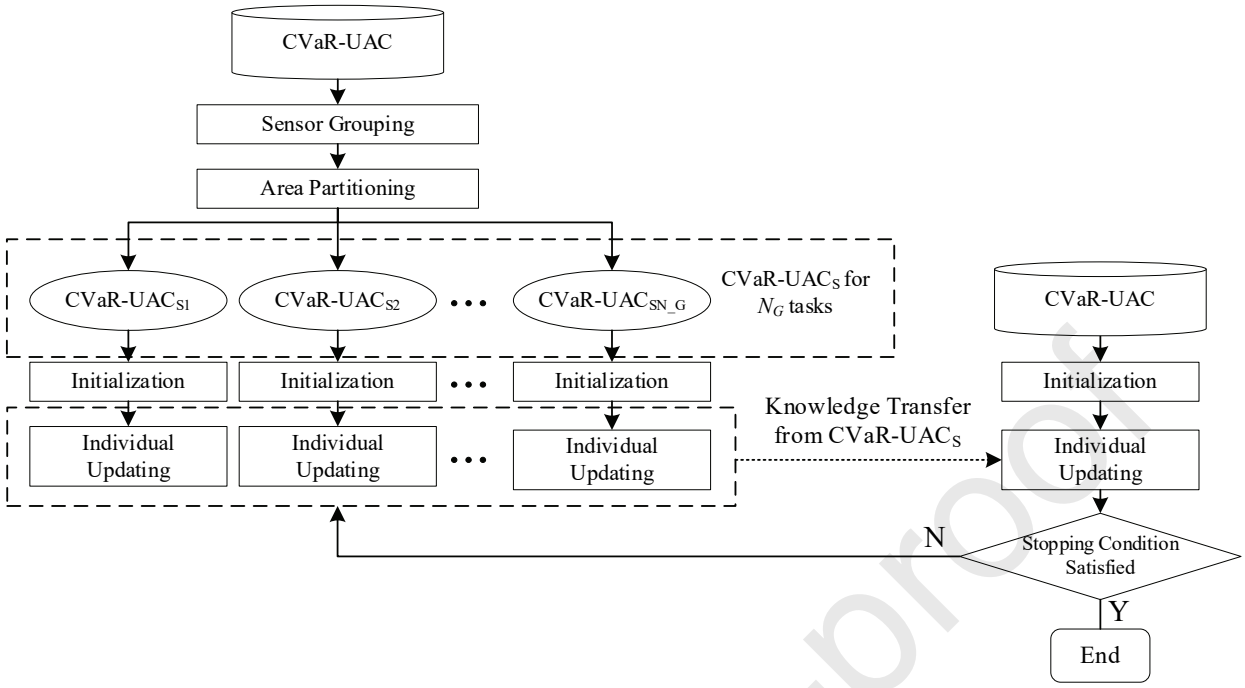


Fig. 4: Illustration of EMT-ad-VFCPSO framework

UAC$_S$ and CVaR-UAC are initialized using a hybrid initialization method. ad-VFCPSO is adopted as the search engine for individual updating. Therefore, the algorithm performs ad-VFCPSO for both the simplified CVaR-UAC$_S$ and the original CVaR-UAC, while knowledge transfer is conducted between them. The pseudo-code of the proposed EMT-ad-VFCPSO is shown in Algorithm 2. The evolutionary process consists of three parts, including the individual updating for CVaR-UAC$_S$ (line 4–14 in Algorithm 2), knowledge transfer from CVaR-UAC$_S$ to CVaR-UAC (line 15–20 in Algorithm 2), and the individual updating for CVaR-UAC (line 21–30 in Algorithm 2). The processes of individual updating for CVaR-UAC$_S$ and CVaR-UAC are similar since they both adopt ad-VFCPSO. The main difference is that the individuals for CVaR-UAC$_S$ are updated by tasks with properties introduced in Section 3.1.

Knowledge transfer is a key issue in EMT, as it helps in generating new individuals during the evolutionary process. Based on the encoding scheme, the individuals for simplified CVaR-UAC$_S$ are used to further improve the individuals for the original CVaR-UAC. Therefore, before the individual updating for CVaR-UAC, the best deployment schemes for each area obtained from different CVaR-UAC$_S$ are combined as a deployment scheme p_g^{emt} for all areas to update the global best particle p_g in CVaR-UAC. Furthermore, similar to the CPSO, each dimension of the global best particle in CVaR-UAC is updated by p_g^{emt} in a co-evolutionary manner. The simplified CVaR-UAC$_S$ is less complex than the original CVaR-UAC. With the help of simplified tasks, the individuals of the CVaR-UAC will converge faster.

3.5. Computational complexity

The computational complexities of the fitness evaluation of the simplified CVaR-UAC$_S$ and CVaR-UAC are $O(A_{search}^{emt} \cdot N^{emt} \cdot W)$ and $O(A_{search} \cdot N \cdot W)$, respectively, where A_{search}^{emt} is the number of grids to be covered in the simplified CVaR-UAC$_S$, A_{search} is the number of grids to be covered in CVaR-UAC, N^{emt} is the total number of sensors in the simplified CVaR-UAC$_S$, N is the total number of sensors in CVaR-UAC, and W is the total number of scenarios for CVaR calculation.

The computational complexities of the calculation of total virtual force for simplified CVaR-UAC$_S$ and CVaR-UAC are $O((N^{emt})^2)$ and $O(N^2)$, respectively.

As for the initialization, the computational complexities with VFA of the simplified CVaR-UAC$_S$ and CVaR-UAC are $O(N_p^{emt} \cdot N_G \cdot MaxLoop \cdot N^{emt} \cdot (A_{search}^{emt} \cdot W + N^{emt}))$ and $O(N_p^{main} \cdot MaxLoop \cdot N \cdot (A_{search} \cdot W + N))$, respectively,

Algorithm 2 EMT-ad-VFCPSO

Input:

N_p^{emt}, N_p^{main}: Particle size for CVaR-UAC$_S$ and CVaR-UAC

$D, task(k).D$: Swarm size (dimension size) for CVaR-UAC and task k of CVaR-UAC$_S$

f, f_k^{emt}: Objective function for CVaR-UAC and CVaR-UAC$_S$ of task k

$b(j,z) \equiv (p_{g1}, p_{g2}, p_{g3}, \ldots, p_{g(j-1)}, z, p_{g(j+1)}, \ldots, p_{gD})$

$task(k).b(j,z) \equiv (task(k).p_{g1}, task(k).p_{g2}, task(k).p_{g3}, \ldots, task(k).p_{g(j-1)}, z, task(k).p_{g(j+1)}, \ldots, task(k).p_{g,task(k).D})$

Output:

p_g^*: Final solution (global best particle)

Construct multiple simplified sensor deployment problems (CVaR-UAC$_S$).
Initial the particles, pg, pc and pl for both individuals for simplified CVaR-UAC$_S$ and CVaR-UAC.
while Stopping criteria not satisfied **do**
    /* Individual updating for CVaR-UAC$_S$ */
    for $k = 1$ to N_G **do**
        for $j = 1$ to $task(k).D$ **do**
            for $i = 1$ to N_p^{emt} **do**
                Determine which disturbance operator to be conducted based on the selection ratio pg, pc, and pl.
                Update particles by Eqs. (11) and (12) for CVaR-UAC$_S$.
                Update $task(k).p_{ij}$ by $f_k^{emt}(task(k).b(j, task(k).x_{ij})$ and $f_k^{emt}(task(k).b(j, task(k).p_{ij})$ and update the successful number of perturbations.
                Update $task(k).p_{gj}$ by $f_k^{emt}(task(k).b(j, task(k).p_{ij})$ and $f_k^{emt}(task(k).p_g)$.
            end for
        end for
    end for
    /* Knowledge transfer from CVaR-UAC$_S$ to CVaR-UAC */
    Construct a deployment scheme p_g^{emt} from the global best particles of all tasks $task.p_g$.
    Update p_g by $min\{f(p_g^{emt}), f(p_g))\}$.
    for $j = 1$ to D **do**
        Update p_{gj} by $min\{f(b(j, p_{gj}^{emt}), f(p_g))\}$.
    end for
    /* Individual updating for CVaR-UAC */
    for $j = 1$ to D **do**
        for $i = 1$ to N_p^{main} **do**
            Determine which disturbance operator to be conducted based on the selection ratio pg, pc, and pl.
            Update particles by Eqs. (11) and (12) for CVaR-UAC.
            Update p_{ij} by $min\{f(b(j, x_{ij}), f(b(j, p_{ij}))\}$ and the successful number of perturbations.
            Update p_{gj} by $min\{f(b(j, p_{ij}), f(p_g))\}$.
        end for
        Update pg, pc, and pl by Eqs. (14), (15), and (16).
    end for
end while
$p_g^* = p_g$.
return

where N_p^{emt} is the population size for the simplified CVaR-UAC$_S$, N_G is the total task groups, N_p^{main} is the population size for the CVaR-UAC, and $MaxLoop$ is the maximum number of loops for VFA.

In each iteration of the evolution, there are three parts, which are the individual updating for CVaR-UAC$_S$, knowledge transfer from CVaR-UAC$_S$ to CVaR-UAC, and the individual updating for CVaR-UAC. The computational complexities for the three parts are $O(N_p^{emt} \cdot N_G \cdot D^{emt} \cdot N^{emt} \cdot (A_{search}^{emt} \cdot W + N^{emt}))$, $O(D \cdot A_{search} \cdot N \cdot W)$, and $O(N_p^{main} \cdot D \cdot N \cdot (A_{search} \cdot W + N))$, respectively, where D^{emt} is the dimension of the simplified CVaR-UAC$_S$ and D is

the dimension of the CVaR-UAC.

To sum up, the computational complexity of EMT-ad-VFCPSO is $O(N_p^{emt} \cdot N_G \cdot N^{emt} \cdot (A_{search}^{emt} \cdot W + N^{emt}) \cdot (MaxLoop + n_{iter} \cdot D^{emt}) + N_p^{main} \cdot N \cdot (A_{search} \cdot W + N) \cdot (MaxLoop + n_{iter} \cdot D))$, where n_{iter} is the number of iterations. Since $A_{search}^{emt} \cdot W \gg N^{emt}$, $n_{iter} \cdot D^{emt} \gg MaxLoop$, $A_{search} \cdot W \gg N$, $n_{iter} \cdot D \gg MaxLoop$, $D^{emt} = 2N^{emt}$, and $D = 2N$, the computational complexity of EMT-ad-VFCPSO is $O(n_{iter} \cdot N_p^{emt} \cdot N_G \cdot A_{search}^{emt} \cdot W \cdot (N^{emt})^2 + n_{iter} \cdot N_p^{main} \cdot A_{search} \cdot W \cdot N^2)$.

Remark 3. *When $N_p^{emt} = 0$ and $N_p^{main} = N_p$, EMT-ad-VFCPSO degenerates to ad-VFCPSO with hybrid initialization without EMT (EMT-ad-VFCPSO-1). The computational complexity of EMT-ad-VFCPSO-1 is $O(n_{iter} \cdot N_p \cdot A_{search} \cdot W \cdot N^2)$. When $N_p^{main} = 0$ and $N_p^{emt} = N_p$, all individuals are used for simplified CVaR-UAC$_S$ (EMT-ad-VFCPSO-2). The computational complexity of EMT-ad-VFCPSO-2 is $O(n_{iter} \cdot N_p \cdot N_G \cdot A_{search}^{emt} \cdot W \cdot (N^{emt})^2)$. Since $N \approx N_G \cdot N^{emt}$, $A_{search} > A_{search}^{emt}$, and $N > N^{emt}$, for $N_G \geqslant 2$, the computational complexity of EMT-ad-VFCPSO-1 is larger than that of EMT-ad-VFCPSO-2. As a result, there will be less computation time for EMT-ad-VFCPSO with more proportion of EMT.*

4. Experimentation design

This section presents the algorithms and the experiment. Firstly, we briefly describe the state-of-the-art algorithms used for comparison. Then, we provide test instances and parameter settings used in our experiment.

4.1. Algorithms for comparison

In order to evaluate the performance of EMT-ad-VFCPSO, the following algorithms are considered for comparison.

- ad-VFCPSO is the basis of EMT-ad-VFCPSO, which uses the adaptive perturbation in the velocity updating in the VFCPSO evolution. The hybrid initialization method and EMT are not considered.
- RPSO [44] uses the resampling, mutation, and small vibration process to PSO.
- CMA-ES (Covariance Matrix Adaptation Evolution Strategy) [19] uses a probability model to obtain new solutions sampled from a multivariate normal distribution.
- GA1 [18] uses a tournament selection, Laplace crossover and arithmetic crossover operators, and Gaussian mutation.
- GA2 [47] uses a random selection, BLA-α crossover operator, and Gaussian mutation.

4.2. Test instances

The parameter settings of the test instances are as follows. We assume that all the sensors are heterogeneous and deployed in a $100 \times 100\text{m}^2$ square area with $L = 100$. The grid size *grid* is 5m. The probabilistic detection model parameters are set as $\alpha_1 = 1$, $\alpha_2 = 0$, $\beta_1 = 1$, $\beta_2 = 0.5$. The total number of scenarios is set to $W = 20$. The β in the CVaR risk measure is set to 0.9. For each scenario, the range detection error r_e with uncertainty is sampled from a uniform distribution in $[0.1r, 0.5r]$.

There are three types of sensors with the sensing range r_1, r_2, and r_3, respectively, where $r_2 = 0.8r_1$ and $r_3 = 0.8r_2$. The number of sensors for each type are N_1, N_2, and N_3, with $N_1 + N_2 + N_3 = N$. τ is referred to as the tightness ratio. There are 20 test instances shown in Table 2. Some parameters of these test instances are adopted from [47], except for those uncertainty-related parameters. It is a new version with some parameters widely used in [18, 46, 47].

Table 2: Test instances.

Instance	r_1	N_1	r_2	N_2	r_3	N_3	N	τ
S1-0.7	14.00	5	11.20	5	8.96	7	17	0.68
S2-0.7	12.00	6	9.60	8	7.68	10	24	0.69
S3-0.7	10.00	8	8.00	12	6.40	16	36	0.70
S4-0.7	8.00	12	6.40	18	5.12	27	57	0.70
S5-0.7	6.00	22	4.80	32	3.84	47	101	0.70
S1-0.8	14.00	5	11.20	6	8.96	10	21	0.80
S2-0.8	12.00	6	9.60	9	7.68	14	29	0.79
S3-0.8	10.00	9	8.00	13	6.40	19	41	0.79
S4-0.8	8.00	14	6.40	20	5.12	29	63	0.78
S5-0.8	6.00	25	4.80	36	3.84	55	116	0.80
S1-0.9	14.00	6	11.20	7	8.96	10	23	0.90
S2-0.9	12.00	7	9.60	11	7.68	14	32	0.89
S3-0.9	10.00	11	8.00	14	6.40	21	46	0.90
S4-0.9	8.00	16	6.40	23	5.12	34	73	0.90
S5-0.9	6.00	28	4.80	41	3.84	61	130	0.90
S1-1.0	14.00	7	11.20	8	8.96	10	25	1.00
S2-1.0	12.00	8	9.60	11	7.68	17	36	1.00
S3-1.0	10.00	12	8.00	16	6.40	23	51	0.99
S4-1.0	8.00	18	6.40	27	5.12	35	80	1.00
S5-1.0	6.00	31	4.80	48	3.84	65	144	1.00

4.3. Parameter settings

The simulation was implemented on an Intel Core i7-7700HQ (2.81 GHz) computer with 16 GB RAM using MATLAB R2018b. Several state-of-the-art algorithms were compared with EMT-ad-VFCPSO. The parameter settings of the compared algorithms were kept the same as the original papers. Table 3 shows the general and algorithm-specific parameter settings.

The particle/population size was set to $N_p = 50$, and the maximum number of fitness evaluations was set to $MaxFEs = 25000$. The independent runs for each algorithm on each instance were set to 20. In EMT-ad-VFCPSO, the number of groups/subareas was set to $N_G = 4$. This algorithm also uses c_0 to control the adaptive disturbance and N_p^{emt}/N_p^{main} to control the proportion of simplified CVaR-UAC$_S$ in EMT. Meanwhile, a hybrid initialization method was applied in EMT-ad-VFCPSO. After the empirical studies in Section 5.3, we adopted the parameter settings, i.e., $c_0 = 0.5$, $N_p^{emt}/N_p^{main} = 25$, and a 100% hybrid initialization rate. In order to divide the sensors into groups, we employed a strategy that minimizes the difference between the total sensing area of the sensors in a group. Thus, we obtained four sensor groups with a similar total area.

Note that the specific parameters of RPSO: threshold value of particle aggregation degree (PAD_{th}) and maximum active value (AV_{max}) were not given in the original paper [44]. We experimented PAD_{th} with 50, 100, 200, and 300, AV_{max} with 2, 5, and 10. We found that by empirical testing, $PAD_{th} = 200$, and $AV_{max} = 2$ provided the best overall performance.

5. Results and discussions

This section presents the experimental results on the test instances introduced in Section 4.2. The performance of EMT-ad-VFCPSO is compared with several state-of-the-art algorithms on several categories. Additionally, the analyses of knowledge transfer and parameter settings are presented.

5.1. Performance comparison of algorithms

5.1.1. CVaR of UACR

To evaluate the quality of the obtained solutions, we use the indicator RPD (relative percentage deviation) for measuring the amount of improvement over the CVaR of UACR. A smaller RPD indicates a better solution for CVaR-

Table 3: Parameter settings for EMT-ad-VFCPSO and comparison algorithms.

General Settings	
Particle/population Size	$N_p = 50$
Termination Condition	$MaxFEs = 25000$
Independent Runs	20
Significance Test	The Wilcoxon rank-sum test at a 0.05 significance level
EMT-ad-VFCPSO	
Particle Size	$N_p^{emt} = N_p^{main} = 25$
Inertia Weight	Linear decreasing from 0.9 to 0.4
Disturbance Factor	$c_0 = 0.5$
Acceleration Constants	$c_1 = c_2 = c_3 = 1$
Parameters of Virtual Force	The same as [43]
Hybrid Initialization Rate	100%
Maximum number of loops for adjusting sensor locations using VFA	$MaxLoop = 20$
Minimum Selection Ratio	$\gamma = 0.05$
Number of Groups/Subareas	$N_G = 4$
Sensor Grouping Strategy	Divided into groups with a minimum difference in total area
Area Partitioning Strategy	Divided into 4 subareas with a $50 \times 50\text{m}^2$ square area
RPSO	
Acceleration Constants	$c_1 = c_2 = 2.05$
Threshold Value of Particle Aggregation Degree	$PAD_{th} = 200$
Maximum Active Value	$AV_{max} = 2$
GA1	
Arithmetic Crossover Rate	0.9
Laplace Crossover Rate	0.1
Mutation Rate	$1/N$
Standard Deviation for Gaussian Mutation	Absolute value of the difference between two parents
GA2	
BLA-α Crossover	$\alpha = 0.5$
Mutation Rate	$0.1/N$
Standard Deviation for Gaussian Mutation	$L/2$

UAC. RPD is calculated as

$$RPD = \frac{f(alg) - f(ref)}{f(ref)} \times 100 \tag{17}$$

where $f(alg)$ is the objective value (CVaR of UACR) obtained by any test algorithms and $f(ref)$ is the reference value for an instance. In this paper, $f(ref)$ is the best solution among the compared algorithms on 20 independent runs.

Table 4 shows the statistical results of EMT-ad-VFCPSO on the CVaR of the UACR and RPD compared with other state-of-the-art algorithms. The data show the mean value and standard deviation of the algorithms in 20 independent runs. The mean value of the algorithm is used for calculating the mean RPD. The best results are indicated in **bold**. Each algorithm is compared by the Wilcoxon rank-sum test with a significance level of 0.05. The rank-sum test result is 1 when one algorithm is superior to another. Otherwise, the rank-sum test result is 0. The *Score* denotes the total score of each algorithm against others. It is the sum of the rank-sum test results on all test instances.

The overall performance of EMT-ad-VFCPSO is significantly better than the compared algorithms, winning 86 cases among the 100 comparisons. This result shows that the hybrid initialization method and EMT positively impact the performance of the ad-VFCPSO framework. Specifically, EMT-ad-VFCPSO performs better on 17 instances, while ad-VFCPSO performs second with a total score of 72. ad-VFCPSO performs better on two instances (S1-0.7 and S2-0.9). It shows good performance with the adaptive disturbance on VFCPSO, which proves its effectiveness in

Table 4: Results of the comparison on the objective value on 20 test instances.

Instance	EMT-ad-VFCPSO		ad-VFCPSO		RPSO [44]		CMA-ES [19]		GA1 [18]		GA2 [47]	
	mean ± std	RPD	mean ± std	RPD	mean ± std	RPD	mean ± std	RPD	mean ± std	RPD	mean ± std	RPD
S1-0.7	0.5604 ± 0.0009	0.25	**0.5599 ± 0.0006**	**0.16**	0.5640 ± 0.0019	0.89	0.5606 ± 0.0010	0.29	0.5730 ± 0.0048	2.50	0.5611 ± 0.0014	0.38
S2-0.7	**0.5269 ± 0.0003**	**0.10**	0.5270 ± 0.0003	0.12	0.5332 ± 0.0025	1.30	0.5299 ± 0.0017	0.67	0.5554 ± 0.0104	5.52	0.5289 ± 0.0014	0.49
S3-0.7	**0.4962 ± 0.0006**	**0.32**	0.4964 ± 0.0008	0.36	0.5041 ± 0.0032	1.91	0.4999 ± 0.0024	1.06	0.5369 ± 0.0181	8.53	0.4988 ± 0.0016	0.84
S4-0.7	**0.4370 ± 0.0008**	**0.35**	0.4392 ± 0.0009	0.85	0.4549 ± 0.0057	4.46	0.4550 ± 0.0049	4.49	0.5211 ± 0.0290	19.67	0.4435 ± 0.0041	1.85
S5-0.7	**0.3722 ± 0.0022**	**0.97**	0.3801 ± 0.0032	3.11	0.3928 ± 0.0059	6.55	0.4018 ± 0.0059	9.01	0.4980 ± 0.0294	35.09	0.3886 ± 0.0085	5.42
S1-0.8	**0.4715 ± 0.0014**	**0.36**	0.4717 ± 0.0013	0.42	0.4775 ± 0.0027	1.65	0.4728 ± 0.0016	0.64	0.5024 ± 0.0173	6.94	0.4744 ± 0.0023	0.98
S2-0.8	**0.4478 ± 0.0005**	**0.21**	0.4486 ± 0.0005	0.38	0.4565 ± 0.0041	2.15	0.4517 ± 0.0024	1.08	0.4935 ± 0.0189	10.43	0.4530 ± 0.0039	1.36
S3-0.8	**0.4309 ± 0.0012**	**0.42**	0.4324 ± 0.0013	0.76	0.4432 ± 0.0031	3.28	0.4327 ± 0.0026	0.84	0.4791 ± 0.0106	11.65	0.4358 ± 0.0035	1.56
S4-0.8	**0.3712 ± 0.0015**	**0.66**	0.3748 ± 0.0025	1.66	0.3957 ± 0.0072	7.32	0.3954 ± 0.0087	7.22	0.4772 ± 0.0252	29.42	0.3822 ± 0.0051	3.66
S5-0.8	**0.2856 ± 0.0033**	**2.07**	0.3046 ± 0.0046	8.85	0.3187 ± 0.0058	13.92	0.3302 ± 0.0100	18.03	0.4409 ± 0.0315	57.59	0.3135 ± 0.0087	12.04
S1-0.9	**0.4077 ± 0.0013**	**0.59**	0.4096 ± 0.0015	1.08	0.4143 ± 0.0017	2.22	0.4084 ± 0.0017	0.78	0.4482 ± 0.0099	10.60	0.4110 ± 0.0029	1.42
S2-0.9	0.3753 ± 0.0023	1.13	**0.3752 ± 0.0018**	**1.10**	0.3875 ± 0.0054	4.44	0.3776 ± 0.0045	1.77	0.4349 ± 0.0257	17.19	0.3804 ± 0.0050	2.50
S3-0.9	**0.3642 ± 0.0011**	**0.65**	0.3658 ± 0.0017	1.10	0.3779 ± 0.0058	4.44	0.3687 ± 0.0048	1.88	0.4296 ± 0.0172	18.71	0.3683 ± 0.0034	1.79
S4-0.9	**0.2946 ± 0.0022**	**1.31**	0.3001 ± 0.0029	3.20	0.3258 ± 0.0060	12.04	0.3208 ± 0.0088	10.32	0.4196 ± 0.0305	44.29	0.3118 ± 0.0080	7.22
S5-0.9	**0.2226 ± 0.0040**	**4.48**	0.2424 ± 0.0064	13.76	0.2540 ± 0.0087	19.21	0.2666 ± 0.0131	25.10	0.3944 ± 0.0311	85.10	0.2523 ± 0.0123	18.42
S1-1.0	**0.3528 ± 0.0024**	**1.20**	0.3530 ± 0.0023	1.25	0.3616 ± 0.0030	3.72	0.3564 ± 0.0030	2.22	0.4082 ± 0.0180	17.08	0.3576 ± 0.0032	2.57
S2-1.0	0.3120 ± 0.0026	2.62	0.3123 ± 0.0024	2.72	0.3254 ± 0.0074	7.02	**0.3103 ± 0.0034**	**2.07**	0.3794 ± 0.0185	24.78	0.3162 ± 0.0046	4.00
S3-1.0	**0.2992 ± 0.0026**	**2.25**	0.3010 ± 0.0028	2.89	0.3139 ± 0.0084	7.28	0.2995 ± 0.0049	2.36	0.3837 ± 0.0257	31.14	0.3080 ± 0.0056	5.26
S4-1.0	**0.2314 ± 0.0033**	**2.31**	0.2426 ± 0.0041	7.23	0.2606 ± 0.0090	15.19	0.2559 ± 0.0079	13.15	0.3744 ± 0.0299	65.54	0.2491 ± 0.0104	10.12
S5-1.0	**0.1588 ± 0.0045**	**5.80**	0.1900 ± 0.0059	26.59	0.1950 ± 0.0101	29.92	0.2083 ± 0.0182	38.81	0.3705 ± 0.0269	146.87	0.1937 ± 0.0099	29.05
Average		**1.40**		3.88		7.45		7.09		32.43		5.54
Score†		**86**		72		24		40		0		46

† There are a total of 100 comparisons in the Wilcoxon rank-sum test for each algorithm. Therefore, the full mark of each algorithm is 100.

being used as the framework for EMT-ad-VFCPSO. GA2 performs third among the algorithms, with a total score of 46, while CMA-ES performs fourth with a total score of 40. CMA-ES performs better on instance S2-1.0.

In terms of RPD, the average values of RPD in Table 4 show that EMT-ad-VFCPSO performs better than other algorithms with a value of 1.40, and ad-VFCPSO ranks second. The RPD of EMT-ad-VFCPSO is below 1 on more than half of the test instances, mainly for the instances before S5-0.8 with S1-0.9 and S3-0.9. It means the relative deviation is within 1%, which shows a good performance of EMT-ad-VFCPSO. For other more complex instances, the worst case is 5.8% on instance S5-1.0 from EMT-ad-VFCPSO. It still outperforms other algorithms significantly, which are all over 25%.

The final deployments of the algorithms (EMT-ad-VFCPSO, ad-VFCPSO, RPSO, CMA-ES, GA1, and GA2) with the best performance after optimization on two instances (S4-0.7 and S5-1.0) are shown in Figs. 5 and 6, respectively, with the best in **bold** in the figure caption. EMT-ad-VFCPSO finds the best deployment for these instances.

5.1.2. Convergence analysis

The convergence curve of the algorithms (EMT-ad-VFCPSO, ad-VFCPSO, RPSO, CMA-ES, GA1, and GA2) on two instances (S4-0.7 and S5-1.0) are shown in Fig. 7. It can be observed that EMT-ad-VFCPSO obtains better solutions from the convergence curve in the figure. It also converges fast at the beginning. The reasons are as follows.

- The hybrid initialization method in EMT-ad-VFCPSO is effective in obtaining high-quality initial solutions. As shown in Fig. 7, the initial point of the convergence curve is much smaller than that of other algorithms. This good start for the evolution significantly improves the exploration of the algorithm.
- EMT-ad-VFCPSO first conducts the individual updating for CVaR-UAC$_S$. Since we decompose CVaR-UAC into simplified CVaR-UAC$_S$, the simplified versions are easier to solve than CVaR-UAC based on the complexity of the problem. As a result, the solutions from simplified CVaR-UAC$_S$ can formulate a useful deployment scheme transferred to the original CVaR-UAC to obtain near-optimal solutions at the beginning stage of the evolution.

Among the other algorithms, although RPSO and CMA-ES converge faster to their optimal value at the beginning stage of the evolution, they are gradually surpassed by ad-VFCPSO and GA2, as shown in Fig. 7. ad-VFCPSO stands out in the compared algorithms, although its convergence is accelerated only in the later stage of the evolution.

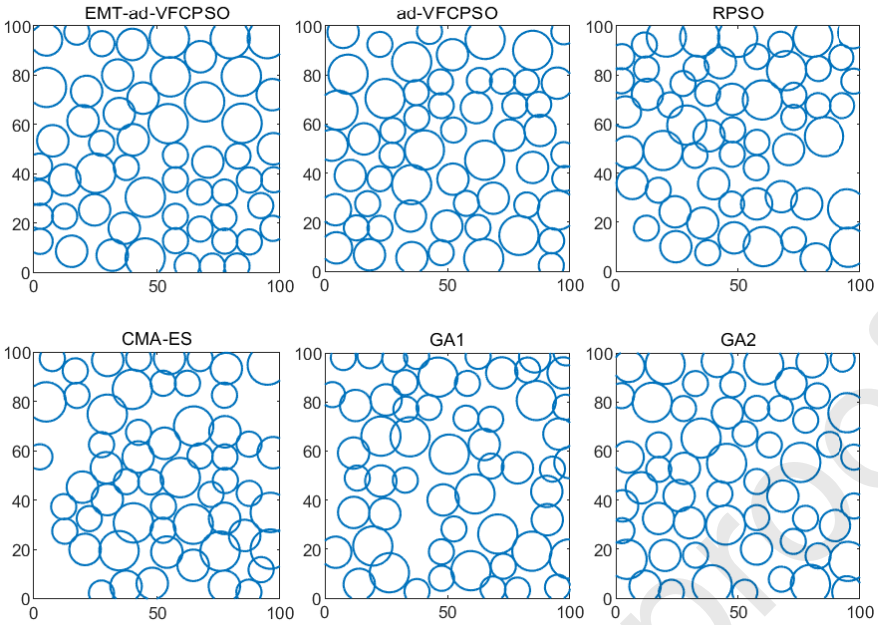


Fig. 5: The best sensor deployments on instance S4-0.7 with objective value (**0.4355**, 0.4377, 0.4460, 0.4454, 0.4800, and 0.4374) in each algorithm.

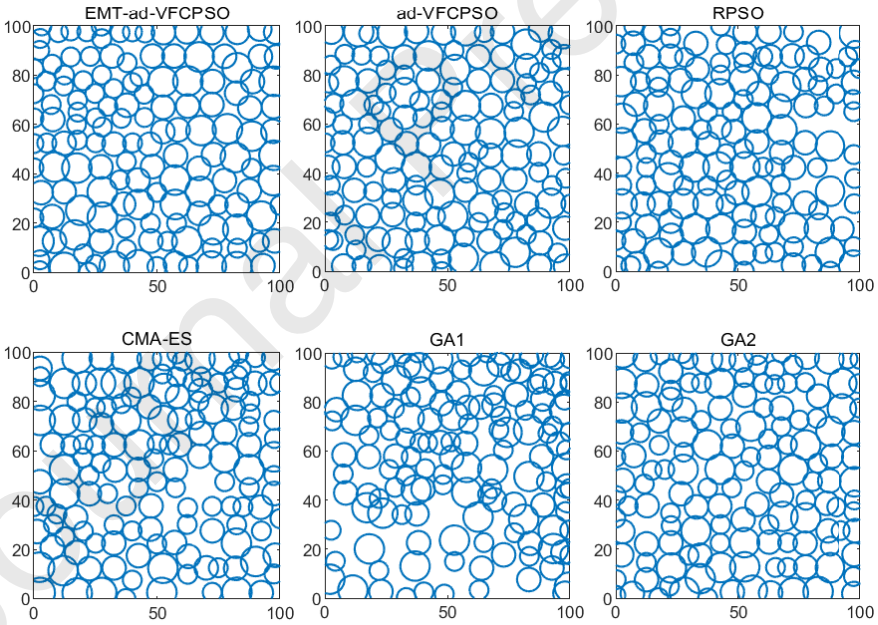


Fig. 6: The best sensor deployments on instance S5-1.0 with objective value (**0.1501**, 0.1824, 0.1752, 0.1793, 0.3063, and 0.1690) in each algorithm.

However, it still cannot exceed EMT-ad-VFCPSO by the end of the evolution. For a more complex instance, S5-1.0, the advantage of EMT-ad-VFCPSO is even more pronounced than the compared algorithms. The superior convergence speed and quality of EMT-ad-VFCPSO demonstrate the efficiency of the proposed ad-VFCPSO, which integrates the hybrid initialization method and EMT, for solving CVaR-UAC.

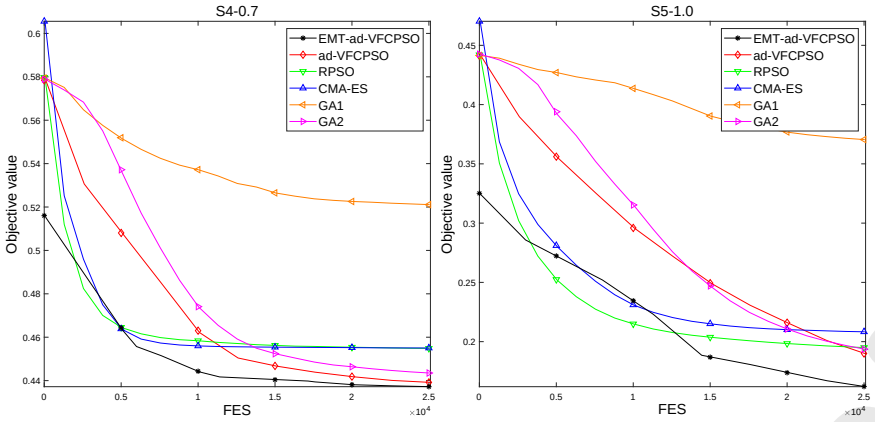


Fig. 7: Convergence analysis of the proposed algorithms on instances (S4-0.7 and S5-1.0).

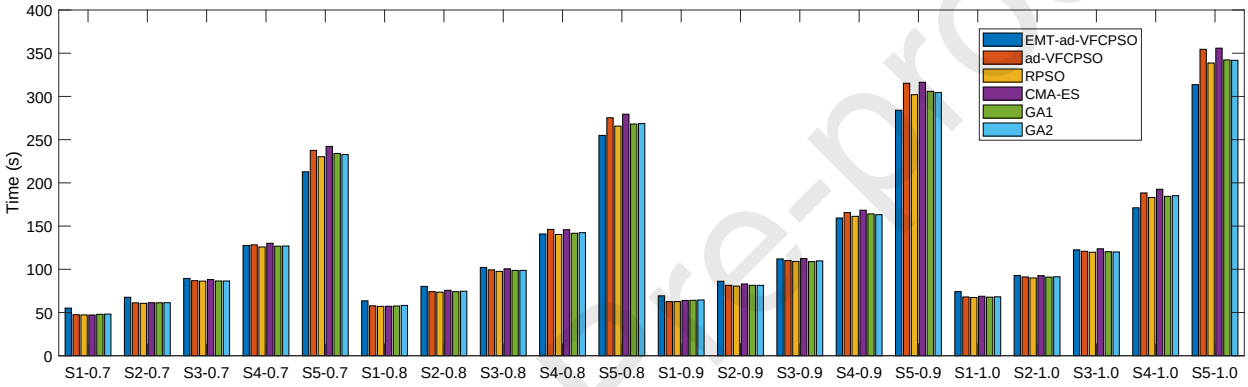


Fig. 8: Average computation time of the proposed algorithms on 20 test instances.

5.1.3. Computation time

Fig. 8 shows the bar graph of the average computation time of the comparison algorithms on 20 test instances. As can be seen, EMT-ad-VFCPSO demonstrates excellent computational efficiency, especially with increased test instances size.

The computation time of EMT-ad-VFCPSO is obviously less than that of other algorithms on instances S5-0.7, S5-0.8, S5-0.9, S4-1.0, and S5-1.0, which are large-scale instances with more sensors to be deployed. The computation time for the fitness evaluations of the simplified CVaR-UAC$_S$ is much less than that of the CVaR-UAC. This indicates that the knowledge transfer in EMT-ad-VFCPSO conserves the computing resources during the evolution process compared with other algorithms.

For small-scale instances, e.g., instances S1-0.7, S2-0.7, S1-0.8, etc., the computation time of EMT-ad-VFCPSO is slightly more than that of other algorithms. This could be attributed to the complex calculation for the hybrid initialization in EMT.

Overall, the proposed EMT-ad-VFCPSO exhibits excellent performance in solution quality, convergence speed, and computation time, demonstrating the effectiveness and efficiency of the proposed efficient PSO with EMT for CVaR-UAC.

5.2. Effectiveness of knowledge transfer

To better show the performance by knowledge transfer in EMT, the illustration of the best transferred solution and the best solution during knowledge transfer from CVaR-UAC$_S$ to CVaR-UAC on 20 instances are shown in Fig. 9. As

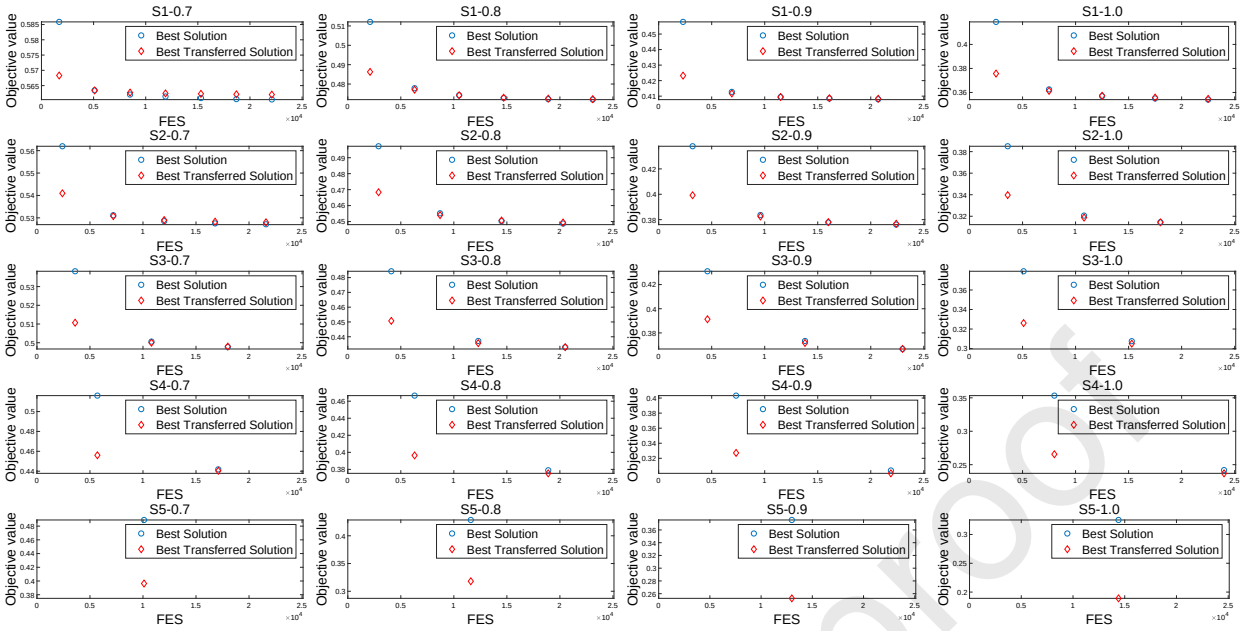


Fig. 9: Illustration of the best transferred solution and the best solution during knowledge transfer from CVaR-UAC$_S$ to CVaR-UAC on 20 test instances.

can be seen, the best transfer solutions are generally better than the best solutions, especially at the beginning of the evolution process. These knowledge transfers have effectively enhanced the search in the evolution process.

The co-evolutionary manner of ad-VFCPSO in the individual updating for CVaR-UAC$_S$ and CVaR-UAC limits the frequency of knowledge transfers. Many fitness evaluations are performed before conducting knowledge transfer. In Fig. 9, the number of knowledge transfers on different instances ranges from 1 to 7, depending on the dimension size of the problem. Selecting an appropriate frequency of knowledge transfer deserves further research, which is beyond the scope of this paper.

5.3. Parameter sensitivity analysis

In this section, we conduct a parameter sensitivity analysis to determine the optimal parameters related to EMT-ad-VFCPSO. The tested parameters are the population size N_p, the adaptive disturbance factor c_0, the hybrid initialization rate for population initialization, and the proportion of simplified CVaR-UAC$_S$ in EMT N_p^{emt}/N_p^{main}. The sensitivity experiments are conducted using EMT-ad-VFCPSO on all test instances with 20 independent runs.

5.3.1. Population size

To analyze the sensitivity of N_p, we test EMT-ad-VFCPSO on all 20 instances, in which $N_p \in \{50, 100, 200\}$. The results are presented in Table 5, where the best results are indicated in **bold**. The mean values for $N_p = 50$ and $N_p = 100$ are the same on instance S2-0.7, but the standard deviation for $N_p = 50$ is smaller, indicating more stable results. All instances for $N_p = 50$ are indicated in **bold**, demonstrating its superiority over $N_p = 100$ and $N_p = 200$. The results in Table 5 are compared by the Wilcoxon rank-sum test with a significance level of 0.05. For EMT-ad-VFCPSO, the best value for N_p is 50, with the highest score of 34. Therefore, in this paper, the particle size N_p in EMT-ad-VFCPSO is set to 50.

5.3.2. Adaptive disturbance factor (c_0)

To analyze the sensitivity of c_0 in EMT-ad-VFCPSO, we test this algorithm on all 20 instances, in which $c_0 \in \{0.1, 0.5, 0.8, 1.0\}$. The results are presented in Table 6, where the best results are indicated in **bold**. The results are compared by the Wilcoxon rank-sum test with a significance level of 0.05. For EMT-ad-VFCPSO, the best value for c_0 is 0.5, with the highest score of 4. Therefore, in this paper, c_0 in EMT-ad-VFCPSO is set to 0.5.

Table 5: Results of the comparison of EMT-ad-VFCPSO with different parameter N_p on 20 test instances.

Instance	$N_p = 50$	$N_p = 100$	$N_p = 200$
S1-0.7	**0.5604 ± 0.0009**	0.5605 ± 0.0008	0.5617 ± 0.0006
S2-0.7	**0.5269 ± 0.0003**	0.5269 ± 0.0004	0.5278 ± 0.0004
S3-0.7	**0.4962 ± 0.0006**	0.4976 ± 0.0006	0.4998 ± 0.0010
S4-0.7	**0.4370 ± 0.0008**	0.4385 ± 0.0011	0.4447 ± 0.0020
S5-0.7	**0.3722 ± 0.0022**	0.3800 ± 0.0021	0.3858 ± 0.0027
S1-0.8	**0.4715 ± 0.0014**	0.4725 ± 0.0013	0.4739 ± 0.0013
S2-0.8	**0.4478 ± 0.0005**	0.4486 ± 0.0007	0.4506 ± 0.0013
S3-0.8	**0.4309 ± 0.0012**	0.4326 ± 0.0017	0.4361 ± 0.0011
S4-0.8	**0.3712 ± 0.0015**	0.3748 ± 0.0022	0.3790 ± 0.0028
S5-0.8	**0.2856 ± 0.0033**	0.2963 ± 0.0036	0.3041 ± 0.0031
S1-0.9	**0.4077 ± 0.0013**	0.4078 ± 0.0009	0.4088 ± 0.0008
S2-0.9	**0.3753 ± 0.0023**	0.3766 ± 0.0018	0.3795 ± 0.0024
S3-0.9	**0.3642 ± 0.0011**	0.3671 ± 0.0013	0.3737 ± 0.0025
S4-0.9	**0.2946 ± 0.0022**	0.3015 ± 0.0030	0.3078 ± 0.0036
S5-0.9	**0.2226 ± 0.0040**	0.2280 ± 0.0046	0.2362 ± 0.0045
S1-1.0	**0.3528 ± 0.0024**	0.3534 ± 0.0025	0.3565 ± 0.0022
S2-1.0	**0.3120 ± 0.0026**	0.3136 ± 0.0031	0.3199 ± 0.0027
S3-1.0	**0.2992 ± 0.0026**	0.3039 ± 0.0034	0.3122 ± 0.0032
S4-1.0	**0.2314 ± 0.0033**	0.2405 ± 0.0033	0.2478 ± 0.0035
S5-1.0	**0.1588 ± 0.0045**	0.1641 ± 0.0053	0.1727 ± 0.0043
Score†	**34**	20	0

† There are a total of 40 comparisons in the Wilcoxon rank-sum test for each algorithm. Therefore, the full mark of each algorithm is 40.

5.3.3. Hybrid initialization rate

Four solution initialization strategies are considered, with hybrid initialization rates (HR) of 0%, 4%, 50%, and 100%, respectively. The notation -HRn represents the variant of EMT-ad-VFCPSO using the initialization strategy with HR $n \in \{0\%, 4\%, 50\%, 100\%\}$. Since the particle size is set to 50 with the value of N_p^{emt} and N_p^{main} are both set to 25. The number of solutions with hybrid initialization is 0, 2 (1 for N_p^{emt} and N_p^{main}), 25 (12 for N_p^{emt} and 13 for N_p^{main}), and 50 (25 for N_p^{emt} and N_p^{main}), respectively.

Table 7 shows the statistical results of EMT-ad-VFCPSO with different initialization strategies. The data show the mean value and standard deviation of the algorithms in 20 independent runs, with the best results indicated in **bold**. Four algorithms are compared by the Wilcoxon rank-sum test with a significance level of 0.05. The total score for -HR100 is 8, which is the best among the algorithms. It can be observed that a higher value of HR yields better performance than a lower HR. Therefore, it is selected for comparison with the state-of-the-art algorithms.

5.3.4. Particle size for EMT

The value of N_p^{emt} and N_p^{main} control the proportion of simplified CVaR-UAC$_S$ in EMT. We consider four combination of N_p^{emt} and N_p^{main} with 25/25, 40/10, 10/40, and 0/50, respectively. Note that for 0/50, there is no particle for simplified CVaR-UAC$_S$, which means EMT is not considered. Table 8 shows the statistical results of EMT-ad-VFCPSO with different particle sizes for CVaR-UAC$_S$ and CVaR-UAC. The data show the mean value and standard deviation of the algorithms in 20 independent runs, with the best results indicated in **bold**. The results are compared by the Wilcoxon rank-sum test with a significance level of 0.05. The total score for the combination 25/25 is 26, which is the best among the combinations. Therefore, the particle sizes for simplified CVaR-UAC$_S$ and CVaR-UAC are set to 25.

Table 6: Results of the comparison of EMT-ad-VFCPSO with different parameter c_0 on 20 test instances.

Instance	$c_0 = 0.1$	$c_0 = 0.5$	$c_0 = 0.8$	$c_0 = 1.0$
S1-0.7	0.5603 ± 0.0010	0.5604 ± 0.0009	0.5603 ± 0.0011	**0.5600 ± 0.0006**
S2-0.7	0.5269 ± 0.0004	0.5269 ± 0.0003	0.5268 ± 0.0004	**0.5268 ± 0.0003**
S3-0.7	0.4965 ± 0.0008	0.4962 ± 0.0006	**0.4962 ± 0.0005**	0.4963 ± 0.0007
S4-0.7	**0.4370 ± 0.0008**	**0.4370 ± 0.0008**	0.4373 ± 0.0012	0.4378 ± 0.0009
S5-0.7	0.3729 ± 0.0023	**0.3722 ± 0.0022**	0.3731 ± 0.0016	0.3731 ± 0.0020
S1-0.8	0.4723 ± 0.0015	**0.4715 ± 0.0014**	0.4718 ± 0.0013	0.4715 ± 0.0016
S2-0.8	0.4481 ± 0.0009	**0.4478 ± 0.0005**	0.4483 ± 0.0010	0.4479 ± 0.0012
S3-0.8	0.4319 ± 0.0012	**0.4309 ± 0.0012**	0.4311 ± 0.0013	0.4312 ± 0.0010
S4-0.8	0.3716 ± 0.0016	**0.3712 ± 0.0015**	0.3719 ± 0.0016	0.3714 ± 0.0013
S5-0.8	0.2872 ± 0.0021	**0.2856 ± 0.0033**	0.2881 ± 0.0031	0.2877 ± 0.0028
S1-0.9	**0.4077 ± 0.0013**	**0.4077 ± 0.0013**	0.4079 ± 0.0011	0.4078 ± 0.0010
S2-0.9	**0.3747 ± 0.0024**	0.3753 ± 0.0023	0.3750 ± 0.0018	0.3749 ± 0.0023
S3-0.9	0.3644 ± 0.0013	**0.3642 ± 0.0011**	0.3649 ± 0.0021	0.3650 ± 0.0016
S4-0.9	0.2933 ± 0.0036	0.2946 ± 0.0022	0.2941 ± 0.0031	**0.2922 ± 0.0029**
S5-0.9	0.2229 ± 0.0026	**0.2226 ± 0.0040**	0.2253 ± 0.0047	0.2233 ± 0.0053
S1-1.0	0.3536 ± 0.0027	0.3528 ± 0.0024	**0.3526 ± 0.0013**	0.3529 ± 0.0020
S2-1.0	0.3117 ± 0.0025	0.3120 ± 0.0026	**0.3108 ± 0.0028**	0.3118 ± 0.0024
S3-1.0	0.2995 ± 0.0021	**0.2992 ± 0.0026**	0.3004 ± 0.0029	0.2997 ± 0.0023
S4-1.0	0.2315 ± 0.0026	**0.2314 ± 0.0033**	0.2325 ± 0.0038	0.2333 ± 0.0039
S5-1.0	**0.1583 ± 0.0060**	0.1588 ± 0.0045	0.1592 ± 0.0040	0.1606 ± 0.0039
Score†	1	**4**	0	1

† There are a total of 60 comparisons in the Wilcoxon rank-sum test for each algorithm. Therefore, the full mark of each algorithm is 60.

5.4. Summary

In summary, the computational results demonstrate that EMT-ad-VFCPSO outperforms other state-of-the-art algorithms. The main reasons are as follows:

- The hybrid initialization method is very effective in obtaining high-quality initial solutions, which significantly enhances the exploration capability of EMT-ad-VFCPSO.
- For a large-scale optimization problem, the co-evolutionary manner and multitasking technique are two efficient approaches. The former employs the divide-and-conquer mechanism to deal with the decision variables, while the latter constructs simplified problems to make the problem easy to solve. As a result, the diversity of the solutions obtained during the search process has been significantly improved.
- The velocity update equation in the original PSO no longer uses the former velocity but instead employs perturbations such as Gaussian, Cauchy, and L'evy functions. These perturbations are stochastic functions that provide more diversity, while using former velocity may lead to an unsatisfying searching direction.
- The disturbance operator adaptively selected from three perturbation operators is better at dealing with the problem than algorithms using a single perturbation. This is because a particular perturbation may not always be effective during the evolution.

6. Conclusion and future work

This paper studies the minimum CVaR-UAC problem, defined as the sensor area coverage problem controlled by CVaR. To generate the initial population, we develop a hybrid initialization method the integrates random and

Table 7: Results of the comparison of EMT-ad-VFCPSO with different initialization on 20 test instances.

Instance	-HR00	-HR04	-HR50	-HR100
S1-0.7	**0.5603 ± 0.0006**	0.5605 ± 0.0011	0.5604 ± 0.0008	0.5604 ± 0.0009
S2-0.7	**0.5268 ± 0.0003**	0.5270 ± 0.0004	0.5270 ± 0.0003	0.5269 ± 0.0003
S3-0.7	0.4964 ± 0.0005	0.4965 ± 0.0006	0.4963 ± 0.0007	**0.4962 ± 0.0006**
S4-0.7	**0.4370 ± 0.0006**	0.4373 ± 0.0012	0.4374 ± 0.0008	0.4370 ± 0.0008
S5-0.7	**0.3720 ± 0.0026**	0.3730 ± 0.0019	0.3732 ± 0.0024	0.3722 ± 0.0022
S1-0.8	0.4727 ± 0.0018	0.4718 ± 0.0018	0.4720 ± 0.0016	**0.4715 ± 0.0014**
S2-0.8	0.4484 ± 0.0011	0.4484 ± 0.0011	0.4481 ± 0.0011	**0.4478 ± 0.0005**
S3-0.8	0.4320 ± 0.0016	0.4316 ± 0.0014	0.4314 ± 0.0013	**0.4309 ± 0.0012**
S4-0.8	0.3721 ± 0.0012	0.3716 ± 0.0018	0.3714 ± 0.0017	**0.3712 ± 0.0015**
S5-0.8	0.2886 ± 0.0036	0.2877 ± 0.0035	0.2872 ± 0.0031	**0.2856 ± 0.0033**
S1-0.9	0.4085 ± 0.0011	0.4085 ± 0.0013	0.4080 ± 0.0012	**0.4077 ± 0.0013**
S2-0.9	0.3762 ± 0.0018	0.3749 ± 0.0018	**0.3746 ± 0.0019**	0.3753 ± 0.0023
S3-0.9	0.3650 ± 0.0016	0.3648 ± 0.0017	0.3657 ± 0.0020	**0.3642 ± 0.0011**
S4-0.9	0.2939 ± 0.0034	**0.2930 ± 0.0029**	0.2933 ± 0.0021	0.2946 ± 0.0022
S5-0.9	0.2237 ± 0.0047	0.2259 ± 0.0040	0.2236 ± 0.0033	**0.2226 ± 0.0040**
S1-1.0	0.3532 ± 0.0016	0.3535 ± 0.0017	0.3531 ± 0.0018	**0.3528 ± 0.0024**
S2-1.0	0.3128 ± 0.0022	0.3129 ± 0.0023	0.3126 ± 0.0028	**0.3120 ± 0.0026**
S3-1.0	0.3002 ± 0.0026	0.3000 ± 0.0044	**0.2991 ± 0.0025**	0.2992 ± 0.0026
S4-1.0	0.2313 ± 0.0047	0.2326 ± 0.0038	**0.2308 ± 0.0032**	0.2314 ± 0.0033
S5-1.0	0.1606 ± 0.0057	0.1603 ± 0.0039	0.1596 ± 0.0068	**0.1588 ± 0.0045**
Score†	0	1	1	**8**

† There are a total of 60 comparisons in the Wilcoxon rank-sum test for each algorithm. Therefore, the full mark of each algorithm is 60.

heuristic strategies. We apply an EMT framework to deal with the problem as simplified helper tasks to obtain new solutions with more diversity. We proposed an efficient PSO based on an adaptive disturbance, a co-evolutionary manner, and virtual force. We provide the approximation ratio of the proposed PSO on the minimum D-CVaR-UAC. The experiment results demonstrate the effectiveness of the proposed algorithms compared with other state-of-the-art algorithms.

In the future, we will further investigate efficient methods to solve the CVaR-UAC with EMT. This includes constructing simplified CVaR-UAC problems with other sensor grouping and area partitioning methods, developing new knowledge transfer techniques, analyzing new search engines, etc. We also aim to model the problem as a multi-objective problem [13] and design a corresponding algorithm in a multi-objective optimization framework. Additionally, we will consider problems that contains uncertain variables without knowing the actual probabilities, which will be of great interest [9]. The proposed algorithm can also be applied to other real-world applications, e.g., deploying unmanned aerial/ground vehicles [10], etc.

CRediT authorship contribution statement

Shuxin Ding: Conceptualization, Data Curation, Investigation, Methodology, Software, Validation, Writing - original draft. **Tao Zhang:** Project administration, Supervision. **Chen Chen:** Supervision, Writing - review & editing. **Yisheng Lv:** Formal analysis. **Bin Xin:** Conceptualization, Supervision, Writing - review & editing. **Zhiming Yuan:** Conceptualization, Writing - review & editing. **Rongsheng Wang:** Methodology, Validation. **Panos M. Pardalos:** Supervision, Writing - review & editing.

Table 8: Results of the comparison of EMT-ad-VFCPSO with different particle size for CVaR-UAC$_S$ and CVaR-UAC on 20 test instances.

Instance	N_p^{emt}/N_p^{main}			
	25/25	40/10	10/40	0/50
S1-0.7	0.5604 ± 0.0009	0.5608 ± 0.0009	0.5602 ± 0.0009	**0.5595 ± 0.0004**
S2-0.7	0.5269 ± 0.0003	0.5272 ± 0.0004	**0.5267 ± 0.0003**	0.5270 ± 0.0005
S3-0.7	0.4962 ± 0.0006	0.4973 ± 0.0008	**0.4961 ± 0.0004**	0.4965 ± 0.0006
S4-0.7	**0.4370 ± 0.0008**	0.4385 ± 0.0012	**0.4370 ± 0.0008**	0.4389 ± 0.0013
S5-0.7	**0.3722 ± 0.0022**	0.3776 ± 0.0029	0.3729 ± 0.0021	0.3792 ± 0.0028
S1-0.8	0.4715 ± 0.0014	0.4727 ± 0.0016	0.4714 ± 0.0010	**0.4711 ± 0.0009**
S2-0.8	**0.4478 ± 0.0005**	0.4491 ± 0.0012	0.4481 ± 0.0009	0.4482 ± 0.0010
S3-0.8	**0.4309 ± 0.0012**	0.4319 ± 0.0012	0.4317 ± 0.0017	0.4319 ± 0.0015
S4-0.8	**0.3712 ± 0.0015**	0.3739 ± 0.0024	0.3718 ± 0.0020	0.3734 ± 0.0013
S5-0.8	**0.2856 ± 0.0033**	0.2923 ± 0.0042	0.2900 ± 0.0035	0.3026 ± 0.0034
S1-0.9	0.4077 ± 0.0013	0.4080 ± 0.0008	**0.4076 ± 0.0012**	0.4084 ± 0.0018
S2-0.9	0.3753 ± 0.0023	0.3765 ± 0.0022	0.3745 ± 0.0016	**0.3743 ± 0.0017**
S3-0.9	0.3642 ± 0.0011	0.3678 ± 0.0023	**0.3637 ± 0.0016**	0.3651 ± 0.0019
S4-0.9	0.2946 ± 0.0022	0.2987 ± 0.0025	**0.2944 ± 0.0034**	0.2996 ± 0.0033
S5-0.9	**0.2226 ± 0.0040**	0.2317 ± 0.0043	0.2258 ± 0.0048	0.2389 ± 0.0034
S1-1.0	**0.3528 ± 0.0024**	0.3539 ± 0.0017	0.3539 ± 0.0018	0.3537 ± 0.0022
S2-1.0	**0.3120 ± 0.0026**	0.3134 ± 0.0030	0.3122 ± 0.0023	0.3128 ± 0.0023
S3-1.0	**0.2992 ± 0.0026**	0.3019 ± 0.0022	0.2997 ± 0.0027	0.3022 ± 0.0032
S4-1.0	0.2314 ± 0.0033	0.2368 ± 0.0034	**0.2306 ± 0.0047**	0.2424 ± 0.0055
S5-1.0	**0.1588 ± 0.0045**	0.1649 ± 0.0049	0.1620 ± 0.0059	0.1880 ± 0.0036
Score†	**26**	4	25	8

† There are a total of 60 comparisons in the Wilcoxon rank-sum test for each algorithm. Therefore, the full mark of each algorithm is 60.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The authors would like to thank the editors and anonymous reviewers for their helpful comments and suggestions on improving the presentation of this paper. This work was supported in part by the National Natural Science Foundation of China under Grants 62203468, 62022015, 62273044, and 62088101, in part by the Young Elite Scientist Sponsorship Program by CAST under Grant 2022QNRC001, in part by the Foundation of China State Railway Group Corporation Limited under Grant L2021G003, and in part by the Foundation of China Academy of Railway Sciences Corporation Limited under Grant 2021YJ043. P. M. Pardalos is partially supported by the Paul and Heidi Brown Preeminent Professorship at ISE (University of Florida, USA), and a Humboldt Research Award (Germany).

References

[1] Ab Aziz, N. A. B., Mohemmed, A. W., Alias, M. Y., 2009. A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram. In: 2009 Int. Conf. Networking, Sens. Control. IEEE, pp. 602–607.

[2] Chowdhury, A., De, D., 2021. Energy-efficient coverage optimization in wireless sensor networks based on voronoi-glowworm swarm optimization-k-means algorithm. Ad Hoc Netw. 122, 102660.

[3] Chu, S.-C., Roddick, J. F., Pan, J.-S., 2005. A parallel particle swarm optimization algorithm with communication strategies. J. Inf. Sci. Eng. 21 (4), 809–818.

[4] Deepa, R., Venkataraman, R., 2021. Enhancing whale optimization algorithm with levy flight for coverage optimization in wireless sensor networks. Comput. Electr. Eng. 94, 107359.

[5] Deif, D. S., Gadallah, Y., 2014. Classification of wireless sensor networks deployment techniques. IEEE Commun. Surv. Tutorials 16 (2), 834–855.

[6] den Bergh, F., Engelbrecht, A. P., 2004. A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. 8 (3), 225–239.

[7] Ding, S., Chen, C., Chen, J., Xin, B., 2014. An improved particle swarm optimization deployment for wireless sensor networks. J. Adv. Comput. Intell. Intell. Inform. 18 (2), 107–112.

[8] Ding, S., Chen, C., Zhang, Q., Xin, B., Pardalos, P. M., 2021. Metaheuristics for resource deployment under uncertainty in complex systems, 1st Edition. CRC Press, Boca Raton FL, USA.

[9] Ding, S., Zhang, Q., Yuan, Z., 2020. An under-approximation for the robust uncertain two-level cooperative set covering problem. In: 2020 59th IEEE Conf. Decis. Control. IEEE, pp. 1152–1157.

[10] Ding, Y., Xin, B., Chen, J., 2021. A review of recent advances in coordination between unmanned aerial and ground vehicles. Unmanned Syst. 9 (2), 97–117.

[11] Erişkin, L., 2021. Point coverage with heterogeneous sensor networks: A robust optimization approach under target location uncertainty. Comput. Netw. 198, 108416.

[12] Farahani, R. Z., Asgari, N., Heidari, N., Hosseininia, M., Goh, M., 2012. Covering problems in facility location: A review. Comput. Ind. Eng. 62 (1), 368–407.

[13] Fei, Z., Li, B., Yang, S., Xing, C., Chen, H., Hanzo, L., 2017. A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems. IEEE Commun. Surv. Tutorials 19 (1), 550–586.

[14] Feng, Y., Feng, L., Kwong, S., Tan, K. C., 2022. A multivariation multifactorial evolutionary algorithm for large-scale multiobjective optimization. IEEE Trans. Evol. Comput. 26 (2), 248–262.

[15] Friedrich, T., Neumann, F., 2015. Maximizing submodular functions under matroid constraints by evolutionary algorithms. Evol. Comput. 23 (4), 543–558.

[16] Gupta, A., Ong, Y.-S., Feng, L., 2016. Multifactorial evolution: toward evolutionary multitasking. IEEE Trans. Evol. Comput. 20 (3), 343–357.

[17] Gupta, A., Zhou, L., Ong, Y.-S., Chen, Z., Hou, Y., 2022. Half a dozen real-world applications of evolutionary multitasking, and more. IEEE Comput. Intell. Mag. 17 (2), 49–66.

[18] Hanh, N. T., Binh, H. T. T., Hoai, N. X., Palaniswami, M. S., 2019. An efficient genetic algorithm for maximizing area coverage in wireless sensor networks. Inf. Sci. 488, 58–75.

[19] Hansen, N., Ostermeier, A., 1996. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In: Proc. IEEE Int. Conf. Evol. Comput. IEEE, pp. 312–317.

[20] Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. In: Proc. ICNN'95 - Int. Conf. Neural Networks. Vol. 4. IEEE, pp. 1942–1948.

[21] Kulkarni, R. V., Venayagamoorthy, G. K., 2011. Particle swarm optimization in wireless-sensor networks: A brief survey. IEEE Trans. Syst., Man, Cybern.–Part C: Appl. Rev. 41 (2), 262–267.

[22] Li, C., Yang, S., Korejo, I., 2008. An adaptive mutation operator for particle swarm optimization. In: Proc. 2008 UK Work. Comput. Intell. IEEE, pp. 165–170.

[23] Li, J.-Y., Zhan, Z.-H., Tan, K. C., Zhang, J., 2022. A meta-knowledge transfer-based differential evolution for multitask optimization. IEEE Trans. Evol. Comput. 26 (4), 719–734.

[24] Li, S., Xu, C., Pan, W., Pan, Y., 2005. Sensor deployment optimization for detecting maneuvering targets. In: 2005 8th Int. Conf. Inf. Fusion. IEEE, pp. 1629–1635.

[25] Li, Y., Gong, W., Li, S., 2023. Multitasking optimization via an adaptive solver multitasking evolutionary framework. Inf. Sci. 630, 688–712.

[26] Liang, C. K., Chung, C. Y., Li, C. F., 2014. A virtual force based movement scheme for area coverage in directional sensor networks. In: 2014 Tenth Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. IEEE, pp. 718–722.

[27] Mantegna, R. N., 1994. Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. Phys. Rev. E 49 (5), 4677–4683.

[28] Nemhauser, G. L., Wolsey, L. A., Fisher, M. L., 1978. An analysis of approximations for maximizing submodular set functions-I. Math. Program. 14 (1), 265–294.

[29] Nguyen, T.-T., Pan, J.-S., Wu, T.-Y., Dao, T.-K., Nguyen, T.-D., 2019. Node coverage optimization strategy based on ions motion optimization. J. Netw. Intell. 4 (1), 1–9.

[30] Ni, Q., Du, H., Pan, Q., Cao, C., Zhai, Y., 2017. An improved dynamic deployment method for wireless sensor network based on multi-swarm particle swarm optimization. Nat. Comput. 16 (1), 5–13.

[31] Qiao, K., Yu, K., Qu, B., Liang, J., Song, H., Yue, C., Lin, H., Tan, K. C., 2022. Dynamic auxiliary task-based evolutionary multitasking for constrained multi-objective optimization. IEEE Trans. Evol. Comput., 1–15.

[32] Rockafellar, R. T., Uryasev, S., 2000. Optimization of conditional value-at-risk. J. Risk 2 (3), 21–42.

[33] Shang, Q., Huang, Y., Wang, Y., Li, M., Feng, L., 2022. Solving vehicle routing problem by memetic search with evolutionary multitasking. Memetic Comput. 14, 31–44.

[34] Sheikh-Hosseini, M., Hashemi, S. R. S., 2022. Connectivity and coverage constrained wireless sensor nodes deployment using steepest descent and genetic algorithms. Expert Syst. Appl. 190, 116164.

[35] Sorokin, A., Boyko, N., Boginski, V., Uryasev, S., Pardalos, P. M., 2009. Mathematical programming techniques for sensor networks. Algorithms 2 (1), 565–581.

[36] Sun, X., Cassandras, C. G., Meng, X., 2019. Exploiting submodularity to quantify near-optimality in multi-agent coverage problems. Automatica 100, 349–359.

[37] Tam, N. T., Dat, V. T., Lan, P. N., Binh, H. T. T., Swami, A., et al., 2021. Multifactorial evolutionary optimization to maximize lifetime of wireless sensor network. Inf. Sci. 576, 355–373.

[38] Tanergüçlü, T., Maraş, H., Gencer, C., Aygüneş, H., 2012. A decision support system for locating weapon and radar positions in stationary point air defence. Inf. Syst. Frontiers 14 (2), 423–444.

[39] Tang, M., Chen, S., Zheng, X., Wang, T., Cao, H., 2018. Sensors deployment optimization in multi-dimensional space based on improved particle swarm optimization algorithm. J. Syst. Eng. Electron. 29 (5), 969–982.

[40] Tohidi, E., Amiri, R., Coutino, M., Gesbert, D., Leus, G., Karbasi, A., 2020. Submodularity in action: From machine learning to signal processing applications. IEEE Signal Process. 37 (5), 120–133.

[41] Wang, H., Wang, W., Wu, Z., 2013. Particle swarm optimization with adaptive mutation for multimodal optimization. Appl. Math. Comput. 221, 296–305.

[42] Wang, X., Wang, S., Ma, J., 2006. Dynamic deployment optimization in wireless sensor networks. In: Intell. Control Autom. Springer, pp. 182–187.

[43] Wang, X., Wang, S., Ma, J.-J., 2007. An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment. Sensors 7 (3), 354–370.

[44] Wang, X., Zhang, H., Fan, S., Gu, H., 2018. Coverage control of sensor networks in IoT based on RPSO. IEEE Internet Things J. 5 (5), 3521–3532.

[45] Wang, Y., Xin, B., Chen, J., 2022. An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources. IEEE Trans. Cybern. 52 (11), 11526–11538.

[46] Yoon, Y., Kim, Y.-H., 2013. An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks. IEEE Trans. Cybern. 43 (5), 1473–1483.

[47] Yoon, Y., Kim, Y.-H., 2022. Maximizing the coverage of sensor deployments using a memetic algorithm and fast coverage estimation. IEEE Trans. Cybern. 52 (7), 6531–6542.

[48] Zhan, Z.-H., Zhang, J., Li, Y., Chung, H. S.-H., 2009. Adaptive particle swarm optimization. IEEE Trans. Syst. Man, Cybern. Part B Cybern. 39 (6), 1362–1381.

[49] Zou, Y., Chakrabarty, K., 2003. Sensor deployment and target localization based on virtual forces. In: INFOCOM 2003. Twenty-Second Annu. Jt. Conf. IEEE Comput. Commun. IEEE Soc. Vol. 2. IEEE, pp. 1293–1303.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Shuxin Ding: Conceptualization, Data Curation, Investigation, Methodology, Software, Validation, Writing - original draft.
Tao Zhang: Project administration, Supervision.
Chen Chen: Supervision, Writing - review & editing.
Yisheng Lv: Formal analysis.
Bin Xin: Conceptualization, Supervision, Writing - review & editing.
Zhiming Yuan: Conceptualization, Writing - review & editing.
Rongsheng Wang: Methodology, Validation.
Panos M. Pardalos: Supervision, Writing - review & editing.