

Target Coverage-optimized Design and Operation of Wireless Visual Sensor Networks

Xiaojian Zhu

School of Computer Science and Engineering
Changshu Institute of Technology
zhuxj5318@163.com

MengChu Zhou

Department of Electrical and Computer Engineering
New Jersey Institute of Technology
zhou@njit.edu

Abstract

Wireless visual sensor networks are deployed and operated to generate, transmit, and process much visual information. Their huge data tend to consume their nodes' much power and time. This work aims to reduce event-reporting delay and maximize network lifespan by 1) properly deploying two types of nodes, i.e., sensor nodes and edge ones; 2) designing optimal data transmission routes; and 3) designing a data offloading strategy. It formulates a multiobjective optimization problem and solves it by proposing and using a differential evolution-based intelligent optimization method. The solution method is compared with a well-known genetic algorithm-based multiobjective optimization one. The simulation results show its effectiveness and superiority over its peer, thus advancing the field of wireless visual sensor networks.

Keywords

Coverage, design, operation, edge computing, Pareto sets, wireless visual sensor networks

1 Introduction

Various cameras [3, 33, 34] are employed as visual sensors to gain images and videos of a variety of objects in wireless visual sensor networks (WVSNs) [1, 19, 32, 41]. They can be widely adopted for target detection, recognition, and tracking and Internet of Behavior [36]. Although their application potential is high, their actual operation often meets two issues: 1) short network lifespan; and 2) unacceptable event-reporting delay (ERD).

WVSNs are designed to collect huge amounts of image/video data, whose generation and transmission take much energy and time of their sensor nodes (SNs). Due to the finite capacities of their batteries, SNs have short lifespan, and so do WVSNs. It becomes an important research topic about how to maximize the lifetime of WVSNs. In

conventional WVSNs, all data are collected by a base station and then sent by it to a remote cloud data center for their processing. This tends to result in an unacceptable ERD. Such ERD indicates low Quality of Experience (QoE) for users. In most monitoring scenes, large ERD means delayed responses to accidents or disasters, thereby causing enormous losses to stakeholders. Hence, reducing it to the smallest is the main concern in designing and operating WVSNs.

Impressed by the decrease of task response delay of users resulted from the deployment of cloudlets close to users in mobile edge/fog computing systems [12, 28, 35], we propose to place a new type of nodes called edge nodes (ENs) close to SNs in a WVSN to handle images to reduce ERD. This results in an edge-enabled WVSN (EWVSN) concept. In it, ENs are responsible for both collecting and processing huge data. Their deployment is expected to significantly reduce data processing delay, relay node count and data routing latency, eventually decreasing ERD drastically in comparison with those without using ENs. Despite such advantages, their adoption can be very costly. This calls for a satisfactory balance between EWVSN's construction cost and user QoE.

Designing an EWVSN is full of challenges because it needs to not only install both SNs and ENs but also assign ENs for SNs. Since cameras have limited sensing angles, we call them directional sensors [38]. To cover targets, some sensing/observation nodes should be placed and their working directions have to be configured. Moreover, for each sensing node, the ENs responsible for collecting and processing its generated data have to be properly designated so as to minimize ERD.

Some studies, such as [4, 20, 22, 23, 25, 30, 41], focus on WVSN deployment optimization. They aim to decrease the WVSN construction cost and enhance the surveillance performance of its concerned areas. In addition to it, the optimization of task scheduling and node configuration is also important. Some researchers pay attention to how to configure nodes and schedule their tasks to improve network performance and area surveillance quality [2, 6, 10, 11, 14, 27, 37].

Different from the prior one, this work studies the optimization of the design and operation of an EWVSN with the objectives to minimize the required SN count, EN count, and network ERD and maximize network lifespan. Note that the four objectives tend to be conflicting. Hence, we aim to make the following contributions: First, we carry out the multiobjective-optimized deployment of an EWVSN for

target-coverage (MDET). Next, we design a multiobjective differential evolution (MODE) algorithm to handle MDET. At last, we compare MODE with the nondominated sorting genetic algorithm III (NSGA-III) through simulations, and show its effectiveness and superiority over NSGA-III.

After Section 2 states the proposed problem, Section 3 details the proposed algorithm for it. We display the simulation results and analyze them in Section 4, and conclude this work in Section 5.

2 Problem Statement

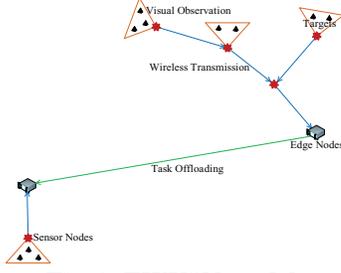


Fig. 1. EWVSN model

An EWVSN is deployed in a 3D scene to cover N^t targets $\{t_i | 1 \leq i \leq N^t\}$, as shown in Fig. 1. As a result, some locations have to be selected from the collection of possible SN placement locations $\{l_i^s | 1 \leq i \leq N^s\}$ to place some observation nodes and some relay nodes. Based on the tasks that it performs, an SN can be only a sensing node, only a relay node, or both ones. Moreover, we need to choose some locations from the collection of possible EN installation locations $\{l_i^e | 1 \leq i \leq N^e\}$ to install some ENs to gather and handle data. Each l_i^s and each l_i^e are predetermined based on various restrictions in the 3D scene. To ease expression, we use t_i to represent its position as well, and use s_i and e_i to denote the SN and EN placed at l_i^s and l_i^e , respectively. We assume all the data sensed by an observation node s_i are not only collected by a single EN but also processed by a single EN. Let $e^c(s_i)$ and $e^p(s_i)$ label the Collection EN (CEN) and Processing EN (PEN) for collecting and processing the data sensed by s_i , respectively. $e^p(s_i)$ may not be $e^c(s_i)$ since an EN is allowed to offload its data processing tasks to another EN to satisfy its computation capacity limit and decrease ERD.

In MDET, besides installing the fewest SNs and the fewest ENs, we should choose observation nodes and their working directions, data transmission paths, CENs, and PENs to cover all targets, maximize network lifetime, and minimize network ERD.

2.1 Target Coverage Model

We use r^s , θ^h , and θ^v to denote the observation radius, horizontal observation angle, and vertical one of each sensing node, respectively. They determine its coverage range modelled as a rectangular pyramid [16, 21, 29], as shown in Fig. 2. We assume that the orientation of every observation node cannot be perpendicular to plane xOy .

In Fig. 2, we assume Q_1Q_2 is parallel to plane xOy at all times. The orientation of s_i intersects rectangle $Q_1Q_2Q_3Q_4$ at Q_5 , and ray s_it_j intersects plane $Q_1Q_2Q_3Q_4$ at Q_7 . Therefore, rectangle $Q_1Q_2Q_3Q_4$ centers at Q_5 . In plane

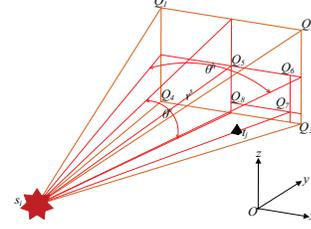


Fig. 2. Visual sensor coverage model

$Q_1Q_2Q_3Q_4$, $Q_5Q_8 \perp Q_5Q_6$, $Q_6Q_7 \perp Q_5Q_6$, and $Q_7Q_8 \parallel Q_5Q_6 \parallel Q_1Q_2$. If $\angle Q_5s_it_j < \pi/2$, $|s_it_j| \leq |s_iQ_7|$, $|Q_5Q_6| \leq r^s \tan(\theta^h/2)$, and $|Q_5Q_8| \leq r^s \tan(\theta^v/2)$, s_i can cover t_j .

The locations where observation nodes can be installed are extracted into $\mathcal{L} = \left\{ i | 1 \leq i \leq N^s \wedge \left(\exists 1 \leq j \leq N^t, \left((r^s \tan(\theta^v/2))^2 + (r^s / \cos(\theta^h/2))^2 \right)^{1/2} \geq |l_i^s t_j| \right) \right\}$.

2.2 Network Lifetime Model

In an EWVSN, ENs are powered by the electrical grid, implying no energy constraints on their operation, but SNs are driven by a battery with finite capacity B . Therefore, the network lifetime depends on that of the latter rather than ENs, and is defined as their shortest lifetime, i.e.,

$$\min\{B/p_i | 1 \leq i \leq N^s \wedge p_i > 0\} \quad (1)$$

where p_i is the energy consumption rate of s_i .

Every SN has a finite transmission radius r^t and a limited transmission rate R . For s_i , the possible SN installation locations to which it can transmit in one hop are collected into $F_i^s = \{j | 1 \leq j \leq N^s \wedge j \neq i \wedge |l_i^s l_j^s| \leq r^t\}$, and the possible EN placement locations to which it can transmit in one hop are collected into $F_i^e = \{j | 1 \leq j \leq N^e \wedge |l_i^s l_j^e| \leq r^t\}$.

If s_i performs the observation task,

$$p_i = p^t \left(\sum_{j \in F_i^s} f_{i,j} + \sum_{j \in F_i^e} f'_{i,j} \right) + p^s \psi \lambda + p^r \sum_{1 \leq j \leq N^s \wedge i \in F_j^s} f_{j,i} \quad (2)$$

otherwise,

$$p_i = p^t \left(\sum_{j \in F_i^s} f_{i,j} + \sum_{j \in F_i^e} f'_{i,j} \right) + p^r \sum_{1 \leq j \leq N^s \wedge i \in F_j^s} f_{j,i} \quad (3)$$

where $f_{i,j}$ and $f'_{i,j}$ are the data traffic via arcs $\langle s_i, s_j \rangle$ and $\langle s_i, e_j \rangle$, respectively, p^s , p^r , and p^t are the energy needed to produce, receive, and transmit a bit data, respectively, ψ is the mean size of an image, and λ is the mean image production speed of an observation node.

2.3 ERD Model

Denote τ_i as an observation node s_i 's ERD, which is the sum of its image routing and processing delay. We define the network ERD as the largest ERD of all observation nodes.

Not only the transmission of images but also their processing can be modelled as an M/M/1 queue [31, 39]. Let $\tau^o(e_i, e_j)$ and P_i label the latency needed to transfer a bit data from e_i to e_j and the node sequence in the data transmission

path of sensing node s_i , respectively. The j th node $P_i(j)$ in P_i is $e^c(s_i)$ if $j = |P_i|$, and an SN otherwise.

The mean sum of waiting and transmission latency of an image at an SN s_i is represented by

$$\tau_i^t = \frac{\Psi}{R - \sum_{j \in F_i^s} f_{i,j} - \sum_{j \in F_i^e} f'_{i,j}} \quad (4)$$

The mean sum of waiting and processing latency of an image at an EN e_i is represented by

$$\tau_i^p = \frac{\eta\Psi}{\xi - \eta\psi\lambda \sum_{j \in \mathcal{L}} y_{i,j}} \quad (5)$$

where ξ is the computation capacity of every EN, η is the mean computing resource requirement of every bit of an image, and $y_{i,j}$ is an indicator being 1 if the images produced by observation node s_j are processed by e_i , and 0 otherwise. If $e^p(s_i) = e^c(s_i)$,

$$\tau_i = \tau_{P_i(|P_i|)}^p + \sum_{1 \leq j \leq |P_i|-1} \tau_{P_i(j)}^t \quad (6)$$

and otherwise,

$$\tau_i = \tau_{e^p(s_i)}^p + \tau^o(e^c(s_i), e^p(s_i))\Psi + \sum_{1 \leq j \leq |P_i|-1} \tau_{P_i(j)}^t \quad (7)$$

MDET is NP-hard because it can be simplified into an NP-hard directional cover set problem in [5].

3 Multiobjective Differential Evolution

3.1 Preliminaries

Differential Evolution (DE) [13, 26] is a widely-used intelligent heuristic which searches for the best solution through mutation, crossover, and selection. Its individuals, denoted by Z_1, Z_2, \dots , and Z_M , are updated generation by generation. In generation g , a mutant one Z_i^g is computed as [26]:

$$Z_i^g = Z_{i_1}^{g-1} + \omega(Z_{i_2}^{g-1} - Z_{i_3}^{g-1}) \quad (8)$$

where any two of i_1, i_2 , and i_3 are different and none of them equals i , and ω is the amplification factor. The j th component of a trial individual Z_i^g is computed by [26]

$$Z_i^g(j) = \begin{cases} Z_i^g(j), & j = j^r \vee \text{rand}() \leq \omega' \\ Z_i^{g-1}(j), & \text{otherwise} \end{cases} \quad (9)$$

where ω' is crossover probability, j^r is randomly selected, and $\text{rand}()$ produces a random number in the range of $[0, 1]$. DE selects the better between Z_i^g and Z_i^{g-1} as Z_i^g .

Our proposed MODE combines DE and the approach to selecting nondominated solutions via reference points in NSGA-III [9, 15] to produce a number of nondominated solutions.

3.2 Individual Representation

Let a 5-tuple $(X^f, X^p, W^x, W^y, W^z)$ represent every individual, where X^f is the parent node index vector, X^p is the PEN vector, W^x, W^y , and W^z are the working direction coordinate vectors.

The i th component $X^f(i)$ of X^f is the index of the parent node of F_i^s inside set F_i^s or F_i^e , and it takes an integer in the

range of $[1 - |F_i^s| - |F_i^e|, |F_i^s| + |F_i^e|]$. Only if $X^f(i) > 0$, s_i is deployed. When s_i is deployed, its parent node is $F_i^s(X^f(i))$ if $X^f(i) \in [1, |F_i^s|]$, and $F_i^e(X^f(i) - |F_i^s|)$ otherwise. The i th component $X^p(i)$ of X^p is $e^p(s_i)$ and takes an integer in $[1 - N^e, 2N^e]$. Only if $X^p(i) > 0$, s_i is an observation node. When s_i is a sensing node, its PEN is $e_{X^p(i)}$ if $X^p(i) \in [1, N^e]$, and the same as its CEN otherwise. $(W^x(i), W^y(i), W^z(i))$ is restricted to be a unit vector and is the working direction of s_i .

3.3 Individual Repair

The target coverage is first repaired. After all targets are covered, the data routes are repaired. Similarly, after the data transmission paths of all observation nodes are successfully built, the adjustment of some observation nodes' PENs is performed.

3.3.1 Target Coverage Repair

We first compute the set of the targets that can be covered by SNs in $S' = \{j | j \in \mathcal{L} \wedge Z.X^p(j) > 0\}$, which use the working directions in $(Z.W^x, Z.W^y, Z.W^z)$. Each time, we greedily select a sensing node from S' , which can increase the most covered targets. We hope that such greedy selection can reduce SNs to be deployed.

If the above selected sensing nodes cannot cover all targets, more sensing nodes need to be selected, which do not use the working directions in $(Z.W^x, Z.W^y, Z.W^z)$. Each time, a sensing node with the largest coverage increment is chosen. However, it is intractable to find an accurate working direction that can maximize the coverage increment. Therefore, a simplified method is proposed to approximately compute it. In this method, for an SN, a new target can be seen as its coverage increment if it can cover both new and old targets.

To judge if s_i is able to cover a target set \mathbb{T} , we should compute 1) projections s_i^h, s_i^v , and s_i^u of s_i and projections $\mathbb{T}^h, \mathbb{T}^v$, and \mathbb{T}^u of \mathbb{T} in planes xOy, xOz , and yOz , respectively; 2) the minimum angles α^h, α^v , and α^u with vertexes s_i^h, s_i^v , and s_i^u that contain all the points in $\mathbb{T}^h, \mathbb{T}^v$, and \mathbb{T}^u , respectively; 3) angles β^h, β^v , and β^u from \vec{Ox} to the bisector vector of α^h , from \vec{Ox} to that of α^v , and from \vec{Oy} to that of α^u , respectively; and 4) the following vector

$$\tilde{w} = \begin{cases} (0, 1, \tan\beta^u), & \beta^u \in [0, 0.5)\pi \cup (1.5, 2)\pi \wedge (\beta^h = (1 \pm 0.5)\pi \vee \beta^v = (1 \pm 0.5)\pi) \\ -(0, 1, \tan\beta^u), & \beta^u \in (0.5, 1.5)\pi \wedge (\beta^h = (1 \pm 0.5)\pi \vee \beta^v = (1 \pm 0.5)\pi) \\ (1, \tan\beta^h, \tan\beta^v), & \beta^v \neq (1 \pm 0.5)\pi \wedge \beta^h \in [0, 0.5)\pi \cup (1.5, 2)\pi \\ -(1, \tan\beta^h, \tan\beta^v), & \beta^v \neq (1 \pm 0.5)\pi \wedge \beta^h \in (0.5, 1.5)\pi \end{cases} \quad (10)$$

s_i can cover \mathbb{T} if \tilde{w} exists and it is able to cover all the targets in \mathbb{T} simultaneously using \tilde{w} as its orientation.

3.3.2 Data Route Repair

To repair $Z.X^f$, the circles in it have to be eliminated first. To connect every observation node to an EN and ensure the transmission rate constraint of every SN, the data routes of observation nodes are iteratively built. Therefore, a graph $G = (V, A, W)$ is constructed based on $Z.X^f$.

$V = \{l^v\} \cup \{l_i^e | 1 \leq i \leq N^e\} \cup \{l_i^s | 1 \leq i \leq N^s\}$ is the vertex collection, where l^v is a virtual vertex used for connecting the possible EN deployment locations. $A = \{\langle l_i^e, l^v \rangle | 1 \leq i \leq N^e\} \cup (\cup_{1 \leq i \leq N^s} A_i^s)$ is the arc set, where the set of the arcs from l_i^s is

$$A_i^s = \begin{cases} \{\langle l_i^s, l_j^e \rangle | j \in F_i^e\} \cup \{\langle l_i^s, l_j^s \rangle | j \in F_i^s\}, & Z.X^f(i) \leq 0 \\ \{\langle l_i^s, F_i^s(Z.X^f(i)) \rangle\}, & Z.X^f(i) \in [1, |F_i^s|] \\ \{\langle l_i^s, F_i^e(Z.X^f(i) - |F_i^s|) \rangle\}, & Z.X^f(i) \geq |F_i^s| + 1 \end{cases} \quad (11)$$

W is the arc weight function. Specially, $W(\langle l_i^s, l_j^s \rangle) = 0$ if $Z.X^f(i) \geq 1$, and 1 otherwise. $W(\langle l_i^s, l_j^e \rangle) = 0$ if $Z.X^f(i) \geq |F_i^s| + 1$, and 1 otherwise. $W(\langle l_i^e, l^v \rangle) = 0$ if there is an arc $\langle l_j^s, l_i^e \rangle$ in A with $W(\langle l_j^s, l_i^e \rangle) = 0$, and $N^s + 1$ otherwise.

For a sensing node s_i , if it is connected to l^v in G , then we adopt the shortest path from it to l^v in G as its data route in $Z.X^f$. As a result of the changes of $Z.X^f$ and data traffic rates at SNs, G has to be rebuilt. Due to the transmission rate restriction of SNs, the arcs that cannot accommodate more data traffic should be deleted from A .

3.3.3 Processing Node Repair

Let $S_i = \{s_j | j \in \mathcal{L} \wedge Z.X^p(j) = i\}$ represent the collection of the observation nodes whose PENs are all e_i . Denote $C = \{e_i | 1 \leq i \leq N^e \wedge \xi \leq \eta \psi \lambda | \{j | j \in \mathcal{L} \wedge Z.X^p(j) = i\}|\}$ and $C' = \{e_i | 1 \leq i \leq N^e \wedge |\{j | j \in \mathcal{L} \wedge Z.X^p(j) = i\}| \in [1, \xi / (\eta \psi \lambda) - 1]\}$ as the collection of the ENs whose workload is not lower than their computing capacity and the collection of the placed ENs which are able to accommodate more workload, respectively. As a result of the computing capacity constraint, the workload of every EN in C must be decreased. If C' is not empty, some tasks of the ENs in C are transferred to the ENs in C' ; otherwise, a new EN is deployed at a randomly selected vacant location and added into C' . For each e_i in C , each time a sensing node is randomly removed from S_i and its PEN is changed to be an EN randomly chosen from C' . If this change makes this EN unable to accommodate more tasks, it has to be deleted from C' . Such operations are repeated until the computing capacity constraint of e_i is met.

After the task offloading in an individual Z is repaired, it is iteratively improved to reduce network ERD. In each iteration, a sensing node s_i with the longest ERD is identified. If we can find a deployed EN e_j that can accommodate the task of processing the data generated by s_i and the network ERD is reduced when it becomes the new PEN of s_i , the workload offloading can be optimized by changing the PEN of s_i to it. If such EN cannot be found, the task offloading cannot be locally optimized further.

3.4 Summary

Given an individual Z , we define its restriction violation degree as:

$$\mathcal{V}(Z) = \begin{cases} \mathcal{V}^e(Z)/N^e, & \mathcal{V}^r(Z) = 0 \wedge \mathcal{V}^s(Z) = 0 \\ \mathcal{V}^s(Z)/N^s(Z) + 1, & \mathcal{V}^r(Z) = 0 \wedge \mathcal{V}^s(Z) > 0 \\ \mathcal{V}^r(Z)/N^r + 2, & \mathcal{V}^r(Z) > 0 \end{cases} \quad (12)$$

where $\mathcal{V}^r(Z)$, $\mathcal{V}^s(Z)$, $\mathcal{V}^e(Z)$, and $N^s(Z) = \sum_{i \in \mathcal{L} \wedge Z.X^p(i) > 0} 1$ are the numbers of uncovered targets, sensing nodes without

data routes, ENs whose computing capacity constraint is violated, and all sensing nodes in Z , respectively. Larger $\mathcal{V}(Z)$ means that Z is worse. Z_i dominates Z_j if 1) $\mathcal{V}(Z_i) < \mathcal{V}(Z_j)$, or 2) $\mathcal{V}(Z_i) = \mathcal{V}(Z_j) = 0$, each objective of Z_i is not worse than that of Z_j , and at least an objective of Z_i is better than that of Z_j .

Before the evolution process starts, the individuals are initialized at random and then repaired. In generation g , to generate a mutant individual Z_i^g , unlike DE, MODE selects two different target individuals $Z_{i_1}^{g-1}$ and $Z_{i_2}^{g-1}$ rather than three, and let

$$Z_i^g = Z_{i_1}^{g-1} + \omega(Z_{i_2}^{g-1} - Z_{i_1}^{g-1}) \quad (13)$$

This modification is expected to better balance exploration and exploitation of our algorithm. After all the individuals in $\{Z_i^g | 1 \leq i \leq M\}$ are produced by (13), (9), and repair, MODE selects M solutions from them and all the individuals in $\{Z_i^{g-1} | 1 \leq i \leq M\}$ to produce $\{Z_i^g | 1 \leq i \leq M\}$. Specially, if we cannot find M feasible solutions in $\{Z_i^g | 1 \leq i \leq M\} \cup \{Z_i^{g-1} | 1 \leq i \leq M\}$, we keep enough solutions with the smallest constraint violation degrees to form M ones; otherwise, we select M ones from all feasible solutions by using the nondominated sorting principle and reference point-based method in NSGA-III [9, 15].

Denote \mathbb{G} as the number of evolution generations. Algorithm 1 realizes MODE.

Algorithm 1 MODE

Input: $l_1^e, l_2^e, \dots, l_{N^e}^e; l_1^s, l_2^s, \dots, l_{N^s}^s; t_1, t_2, \dots, t_{N^r}$

Output: a nondominated solution set

- 1: Generate initial target individuals Z_1^0, Z_2^0, \dots , and Z_M^0 randomly;
 - 2: Repair Z_1^0, Z_2^0, \dots , and Z_M^0 ;
 - 3: **for** $g \leftarrow 1$ to \mathbb{G} **do**
 - 4: **for** $i \leftarrow 1$ to M **do**
 - 5: Compute Z_i^g using (13);
 - 6: Round every $Z_i^g.X^f(j)$ ($1 \leq j \leq N^e$) to its nearest integer;
 - 7: Round every $Z_i^g.X^p(j)$ ($j \in \mathcal{L}$) to its nearest integer;
 - 8: Confine every component of Z_i^g to its value range;
 - 9: Compute Z_i^g using (9);
 - 10: Repair Z_i^g ;
 - 11: **end for**
 - 12: Generate Z_1^g, Z_2^g, \dots , and Z_M^g by choosing M solutions from $\{Z_1^{g-1}, Z_2^{g-1}, \dots, Z_M^{g-1}\} \cup \{Z_1^g, Z_2^g, \dots, Z_M^g\}$;
 - 13: **end for**
 - 14: **return** the set of the different nondominated solutions in $\{Z_1^G, Z_2^G, \dots, Z_M^G\}$;
-

4 Simulations

We compare MODE with NSGA-III [9, 15] in terms of hypervolume (HV) [18, 24, 40], inverted generational distance (IGD) [17, 18, 40], and running time (RT).

Same as in [9, 15], NSGA-III uses the simulated binary crossover (SBX) [8] and polynomial mutation [7]. In addition, it adopts the same individual representation, individual repair, and individual constraint violation degree definition as MODE to handle MDET.

The HV and IGD of a Pareto front indicate the volume of the objective space dominated by it [18, 24, 40] and the mean distance between the points in the true one and their closest

points in it [17, 18, 40], respectively. The accuracy of a Pareto front is in proportion to its HV but in inverse one to its IGD.

For the sake of calculating HV and IGD, we use the shortest network lifetime, longest network lifetime, lowest network ERD, highest network ERD, smallest deployed SN count, largest deployed SN count, smallest deployed EN count, and largest deployed EN count of all the valid solutions that the two algorithms find to normalize the objectives of each nondominated solution output by each algorithm. However, an objective is normalized to 1 if its found minimum and maximum are the same. The Pareto set generated from the two algorithms' results is deemed as the true Pareto set to calculate IGD, and $(-1/N^t, -1/N^t, -1/N^t, -1/N^t)$ is used as the reference point to calculate HV.

4.1 Parameter Settings

The parameters are set in Table 1, where κ is the division count per objective in the approach to selecting nondominated solutions via reference points, ζ and ζ' are the distribution index for crossover and its probability in SBX, respectively, ζ'' and ζ''' are the distribution index for mutation and its probability in the polynomial mutation, respectively. Both algorithms use the same individual count, evolution generation count, and division count per objective in the approach to selecting nondominated solutions via reference points.

Table 1. Parameter Settings

Parameter	Value	Parameter	Value
N^s	250	N^e	20
θ^h	0.6π	ψ	20 Kb/image
λ	10 images/s	η	3000 CPU cycles/b
θ^v	0.4π	r^s	20 m
ξ	3 GHz	p^s	50 nJ/b
p^r	50 nJ/b	p^t	55 nJ/b
B	2000 J	R	1 Mb/s
κ	8	M	40
\mathbb{G}	400	ϖ	0.5
ϖ'	0.3	ζ	30 [9, 15]
ζ'	1 [9, 15]	ζ''	20 [9, 15]
ζ'''	$1/(N^s + 4 \mathcal{L})$ [9, 15]		

HVs, IGDs, and RT for 10 randomly generated instances are averaged. In each instance, all l_i^s , l_i^e , and t_i are generated at random in a $100\text{ m} \times 100\text{ m} \times 50\text{ m}$ 3D space, and every $\tau^o(e_i, e_j)$ is generated in $[0.5, 5]\ \mu\text{s}$ at random.

4.2 Results and Their Analyses

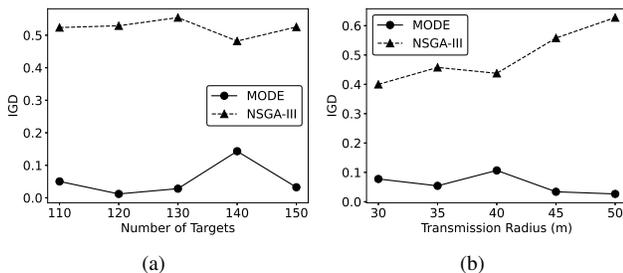


Fig. 3. Evaluation of IGD. (a) IGD versus N^t with $r^t = 40$ m. (b) IGD versus r^t with $N^t = 130$.

Figs. 3-5 display the simulation results. Overall, compared with NSGA-III, MODE takes more time but has larger

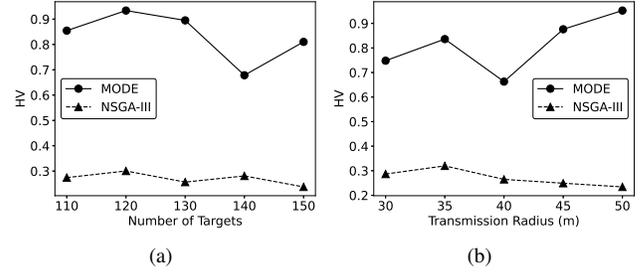


Fig. 4. Evaluation of HV. (a) HV versus N^t with $r^t = 40$ m. (b) HV versus r^t with $N^t = 130$.

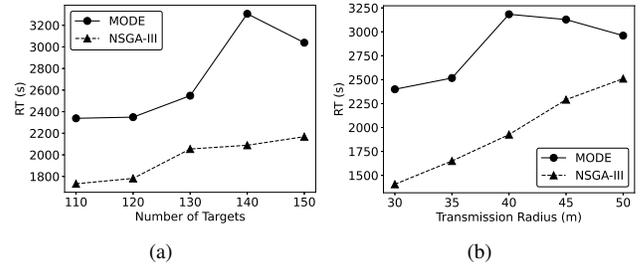


Fig. 5. Evaluation of RT. (a) RT versus N^t with $r^t = 40$ m. (b) RT versus r^t with $N^t = 130$.

HV and smaller IGD. This because MODE is able to better balance exploitation and exploration and discover more feasible solutions during its execution.

The HV of MODE tends to decrease with a few exceptions when the target count rises, as displayed in Fig. 4(a). This is because more observation nodes have to be installed for covering all targets, resulting in the generation of more data, installation of more relay nodes and ENs, shorter network lifespan, and higher network ERD. As the transmission radius of SNs grows, fewer relay nodes are required to be placed. Thus, the network ERD decreases, and the network lifespan rises. Therefore, the HV of MODE tends to increase with a few exceptions, as shown in Fig. 4(b).

As displayed in Fig. 5(a), the algorithms tend to take more time with a few exceptions when the target count increases. This is because larger target count causes bigger problem size. With the increase of the transmission radius of SNs, more feasible solutions can be found, and thereby it consumes more time to repair an individual. As a result, the algorithms tend to consume more time with a few exceptions, as shown in Fig. 5(b).

5 Conclusion

Motivated by the advantages of deploying ENs in WWSNs, we have investigated the optimal deployment of EWWSNs for target coverage with four conflicting objectives. It is hard to address the joint optimization of the deployment of SNs and ENs, and the working direction configuration, data routing, and data offloading of sensing nodes. This work presents a novel algorithm to solve it. By simulations, we discover that it can produce better solutions than NSGA-III at the expense of some additional RT.

6 Acknowledgments

This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant BK20160411.

7 References

- [1] K. Abas, C. Porto, and K. Obraczka. Wireless smart camera networks for the surveillance of public spaces. *Computer*, 47(5):37–44, 2014.
- [2] H. S. Aghdasi and M. Abbaspour. Energy efficient area coverage by evolutionary camera node scheduling algorithms in visual sensor networks. *Soft Computing*, 20(3):1191–1202, 2016.
- [3] I. Ahmed, S. Din, G. Jeon, F. Piccialli, and G. Fortino. Towards collaborative robotics in top view surveillance: A framework for multiple object tracking by detection using deep learning. *IEEE/CAA Journal of Automatica Sinica*, 8(7):1253–1270, 2021.
- [4] A. Ali and H. S. Hassanein. Optimal placement of camera wireless sensors in greenhouses. In *ICC 2021 - IEEE International Conference on Communications*, pages 1–6, 2021.
- [5] Y. Cai, W. Lou, M. Li, and X.-Y. Li. Energy efficient target-oriented scheduling in directional sensor networks. *IEEE Transactions on Computers*, 58(9):1259–1274, 2009.
- [6] R. Dai, P. Wang, and I. F. Akyildiz. Correlation-aware QoS routing with differential coding for wireless video sensor networks. *IEEE Transactions on Multimedia*, 14(5):1469–1479, 2012.
- [7] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, U.K., 2001.
- [8] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995.
- [9] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.
- [10] B. Dieber, C. Micheloni, and B. Rinner. Resource-aware coverage and task assignment in visual sensor networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(10):1424–1437, 2011.
- [11] E. Eriksson, G. Dán, and V. Fodor. Coordinating distributed algorithms for feature extraction offloading in multi-camera visual sensor networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3288–3299, 2018.
- [12] Q. Fan and N. Ansari. On cost aware cloudlet placement for mobile edge computing. *IEEE/CAA Journal of Automatica Sinica*, 6(4):926–937, 2019.
- [13] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou. Chaotic local search-based differential evolution algorithms for optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(6):3954–3967, 2021.
- [14] R. Ghazalian, A. Aghagholzadeh, and S. M. H. Andargoli. Energy optimization and QoE satisfaction for wireless visual sensor networks in multi target tracking scenario. *IEEE Transactions on Multimedia*, 23:823–834, 2021.
- [15] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014.
- [16] F. Jiang, X. Zhang, X. Chen, and Y. Fang. Distributed optimization of visual sensor networks for coverage of a large-scale 3-D scene. *IEEE/ASME Transactions on Mechatronics*, 25(6):2777–2788, 2020.
- [17] Q. Kang, S. Feng, M. Zhou, A. C. Ammari, and K. Sedraoui. Optimal load scheduling of plug-in hybrid electric vehicles via weight-aggregation multi-objective evolutionary algorithms. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2557–2568, 2017.
- [18] Q. Kang, X. Song, M. Zhou, and L. Li. A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(12):2416–2423, 2019.
- [19] A. V. Katsenou, L. P. Kondi, and K. E. Parsopoulos. Motion-related resource allocation in dynamic wireless visual sensor network environments. *IEEE Transactions on Image Processing*, 23(1):56–68, 2014.
- [20] S. Lee, S. Y. Jang, S. J. Hyun, and D. Lee. DQN-based coverage maximization for mobile video camera networks. In *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, pages 01–02, 2021.
- [21] H. Ma, X. Zhang, and A. Ming. A coverage-enhancing method for 3D directional sensor networks. In *IEEE INFOCOM 2009*, pages 2791–2795, 2009.
- [22] T. T. Nguyen, H. D. Thanh, L. Hoang Son, and V. Trong Le. Optimization for the sensor placement problem in 3D environments. In *2015 IEEE 12th International Conference on Networking, Sensing and Control*, pages 327–333, 2015.
- [23] M. Rebai, M. le Berre, F. Hnaïen, and H. Snoussi. Exact biobjective optimization methods for camera coverage problem in three-dimensional areas. *IEEE Sensors Journal*, 16(9):3323–3331, 2016.
- [24] L. M. S. Russo and A. P. Francisco. Quick hypervolume. *IEEE Transactions on Evolutionary Computation*, 18(4):481–502, 2014.
- [25] A. Saad, M. R. Senouci, and O. Benyattou. Toward a realistic approach for the deployment of 3D wireless sensor networks. *IEEE Transactions on Mobile Computing*, 21(4):1508–1519, 2022.
- [26] R. Storn and K. Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
- [27] M. S. S. Suresh, A. Narayanan, and V. Menon. Maximizing camera coverage in multicamera surveillance networks. *IEEE Sensors Journal*, 20(17):10170–10178, 2020.
- [28] V.-T. Truong, V. N. Vo, D.-B. Ha, and C. So-In. On the system performance of mobile edge computing in an uplink NOMA WSN with a multiantenna access point over nakagami- m fading. *IEEE/CAA Journal of Automatica Sinica*, 9(4):668–685, 2022.
- [29] W. Wang, H. Dai, C. Dong, X. Cheng, X. Wang, P. Yang, G. Chen, and W. Dou. Placement of unmanned aerial vehicles for directional coverage in 3D space. *IEEE/ACM Transactions on Networking*, 28(2):888–901, 2020.
- [30] X. Wang, H. Zhang, and H. Gu. Solving optimal camera placement problems in IoT using LH-RPSO. *IEEE Access*, 8:40881–40891, 2020.
- [31] Y. Wang, X. Lin, and M. Pedram. A nested two stage game-based optimization framework in mobile cloud computing system. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, pages 494–502, 2013.
- [32] C. Yu and G. Sharma. Camera scheduling and energy allocation for lifetime maximization in user-centric visual sensor networks. *IEEE Transactions on Image Processing*, 19(8):2042–2055, 2010.
- [33] H. Zhang, L. Jin, and C. Ye. An RGB-D camera based visual positioning system for assistive navigation by a robotic navigation aid. *IEEE/CAA Journal of Automatica Sinica*, 8(8):1389–1400, 2021.
- [34] L. Zhang, H. Han, M. Zhou, Y. Al-Turki, and A. Abusorrah. An improved discriminative model prediction approach to real-time tracking of objects with camera as sensors. *IEEE Sensors Journal*, 21(15):17308–17317, 2021.
- [35] P. Zhang, M. Zhou, and G. Fortino. Security and trust issues in fog computing: A survey. *Future Generation Computer Systems*, 88:16–27, 2018.
- [36] Q. Zhao, G. Li, J. Cai, M. Zhou, and L. Feng. A tutorial on internet of behaviors: Concept, architecture, technology, applications, and challenges. *IEEE Communications Surveys & Tutorials*, 25(2):1227–1260, 2023.
- [37] X. Zhu. Lifetime maximization of connected differentiated target coverage in energy harvesting directional sensor networks. In *2016 IEEE Online Conference on Green Communications (OnlineGreenComm)*, pages 21–26, 2016.
- [38] X. Zhu, J. Li, X. Chen, and M. Zhou. Minimum cost deployment of heterogeneous directional sensor networks for differentiated target coverage. *IEEE Sensors Journal*, 17(15):4938–4952, 2017.
- [39] X. Zhu, E. Setton, and B. Girod. Congestion-distortion optimized video transmission over ad hoc networks. *Signal Processing: Image Communication*, 20(8):773–783, 2005.
- [40] X. Zhu and M. Zhou. Multiobjective optimized cloudlet deployment and task offloading for mobile-edge computing. *IEEE Internet of Things Journal*, 8(20):15582–15595, 2021.
- [41] X. Zhu, M. Zhou, and A. Abusorrah. Optimizing node deployment in rechargeable camera sensor networks for full-view coverage. *IEEE Internet of Things Journal*, 9(13):11396–11407, 2022.