 doi:10.20944/preprints202106.0002.v1

Article

Energy-Efficient Adaptive Sensing Scheduling in Wireless Sensor Networks using Fibonacci Tree Optimization Algorithm

Liangshun Wu [1,†] and Hengjin Cai [2,†,*]

[1] School of Computer Science, Wuhan University; wuliangshun@whu.edu.cn
[2] School of Computer Science, Wuhan University; hjcai@whu.edu.cn
* Correspondence: hjcai@whu.edu.cn; Tel.: +86-180-6205-8525
† This work was supported in part by the National Natural Science Foundation of China under Grant 61832014

Abstract: Wireless sensor networks are attractive largely because they need no wired infrastructure. But precisely this feature makes them energy constrained. Recent studies find that sensing behaviors that are otherwise deemed efficient consume comparable energy with communication. The duty cycle scheduling is perceived as contributing to achieving energy efficiency of sensing. Because of different research assumptions and objectives, various scheduling schemes have various emphases. This paper designed an adaptive sensing scheduling strategy. The objective function of the scheduling strategy includes minimizing average energy expenditure and maximizing sensing coverage (reducing event miss-rate), and it requires relatively loose assumptions. We determine the functional relationship between the variables of the objective function and the step-size parameters of the proposed strategy through the numerical fitting. We found that the objective function aggregated by the fitting functions is a bivariate multi-peak function that favors the Fibonacci tree optimization algorithm. Once the optimization of parameters is done, the strategy can be easily deployed and behaves consistently in the coming hours. We name the proposed strategy as "FTOS". The experimental results show that the Fibonacci tree optimization algorithm gets a better optimistic effect than the comprehensive learning particle swarm optimization (CLPSO) algorithm and differential evolution (DE) algorithm. The FTOS strategy is superior to the fixed time scheduling strategy in achieving the scheduling objectives. It also outperforms other strategies with the same scheduling objectives such as LDAS, BS, DSS and PECAS.

Keywords: sensing; energy-saving; duty cycles; Fibonacci tree optimization

1. Introduction

Wireless Sensor Network (WSN) is a network of thousands of low-cost miniature devices capable of computing, communicating wirelessly, and sensing that runs on a limited battery. Since we typically expect WSN sensors to last from several months to one year without recharging, energy efficiency is essential. Much previous work has referred to energy-efficient communication (MAC and routing) in WSN ([1–4], for example); energy-efficient sensing, however, has not received much attention. Results of recent measurements indicate that sensing uses comparable amounts of energy with wireless communication [5]. Additionally, sensing frequency is higher than communication frequency — for example, a sensor that monitors forest fires only sends a signal when temperatures exceed a threshold. Therefore, it is high time we called for efforts to study energy-saving strategies of sensing.

For a sensing unit, static events, such as temperature and humidity, are not difficult to capture; dynamic events are pretty uncooperative (targets move so much) and therefore difficult to detect, which can only be observed by continuous sensing. Programming sensors' work/sleep cycles (or duty cycles) is considered an useful approach to reduce the energy expenditure caused by continuous sensing. When the node is in sleep mode,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 June 2021 doi:10.20944/preprints202106.0002.v1

only the low-power timer remains active should it be necessary to wake it up. Therefore, the energy expenditure during the sleep interval is only a tiny fraction of that in the working interval.

Although many studies have investigated sensing scheduling problems, they differ in their assumptions and design objectives, and the business requirements of the applications determine them. Based on the beliefs of minimizing event miss-rate and energy expenditure, we strive to design an energy-efficient adaptive sensing scheduling strategy with step-size parameters optimized by Fibonacci tree optimization (abbreviated "FTOS") for wireless sensor networks that would be better than the studies before.

The rest of the paper proceeds: section 2 reviews related work; section 3 introduces the system model; section 4 determines the model parameters through the Fibonacci tree optimization algorithm; section 5 explains why the Fibonacci tree optimization algorithm is chosen for parameter optimization and analyzes the advantages of the proposed FTOS strategy; section 6 implements experiments; section 7 discusses the deficiencies and unresolved issues. Finally, section 8 draws the conclusion.

2. Related Work

2.1. Sensing scheduling strategies

The researches on energy-saving sensing scheduling strategies have two levels. The first level is adaptive duty cycle scheduling. The duty cycle includes active time, nap time, and idle listening period of the detectors; the goal is to best model the event arrival patterns. The second level, coordinated/collaborative sensing. It considers the sensing coverage: how numerous sensor nodes (not just a single node) with different spatiotemporal coordinates in a public area can cooperate to achieve adequate coverage, which is conducive to global optimization of energy consumption.

We know sensors consume the most energy when they are on duty and the least when they are sleeping. Therefore, almost all scheduling strategies, such as ELECTION [6] and AIDS [7], make full use of the energy-saving characteristic of the sleep mode. DANCE [8] improves AIDS by incorporating the behavior of neighboring nodes into consideration scope. In DANCE, the sensor abandons its task if its neighbor has already performed it, thus down-scaling the energy inefficiency. The authors claim that the data sampling rate affects the computing and communication load on the central server. Controlling the sampling rate within a specific range (if it exceeds, then the central server and nodes renegotiate a range) through Kalman filtering can solve this problem [9].

Jothiraj *et al.* [10] recommends fine-tuning the sensing frequency of each sensor dynamically. As the data increases, the accuracy of the sensing improves. The difficulty, however, lies in modeling the actual world.

Maintaining the most significant possible coverage is a primary goal when designing a sensor network. Chen *et al.* [11] defined a sensing coverage metric that measures a wireless sensor network's QoS (quality of service). Besides, an optimization algorithm of polynomial-time complexity using graphing theory and computational geometry is proposed to achieve optimal coverage. [12] extends [11] by algorithm improvement. The first work that considers both energy consumption and sensing coverage was made by [13], introducing an Energy-efficient Surveillance System (ESS). Other studies include LDAS [14,15], PEAS [16], PECAS [17], RIS [18], etc. They make the following two common assumptions: 1) each sensor is power-constrained, and 2) they expect the network to run for a long time. Other assumptions involve:

- *Network Structure*. The network structure can be flat. Each node is homogeneous with the same roles and functionalities. It can also be hierarchical, viz., some nodes can act as the fusion centers.
- *Sensor Placement*. The sensing coverage is usually affected by how the sensors are initially placed. In most cases, the sensors follow a random distribution; there are also studies assuming that sensors obey a two-dimensional (2-D) Poisson distribution [12,33].

- *Sensing Area*: The sensing area can be 2-D circular or 3-D spherical.
- *Time Synchronization*. The sensors are synchronized in time, which means that they can wake up simultaneously for the next scheduling round.
- *Failure Model*. Almost all the studies assume sensors fail when energy is exhausted.
- *Sensor Mobility*. Few studies have addressed this feature, but some have assumed that the sensor is immobile.
- *Location Information*. These studies typically associate location information with whether or how much a sensor's sensing area overlaps with its neighbor's.
- *Distance Information*. The distance information can be inferred from the location information.

Randomized Independent Scheduling (RIS) [18] can extend the sensor lifetime and get an asymptotic k-coverage. And more importantly, it's simple. RIS does not need to provide location information or distance information, nor does it need adjustable transmission range and mobility. However, it has strict distribution assumptions — for example, the sensor is Poisson/uniform/grid distributed, the sensing range follows a uniform distribution, the network is flat and two-dimensional.

LDAS [15], PEAS [16], PECAS [17], etc., have relatively loose assumptions. LDAS assumes that the sensor nodes do not own a positioning device, such as GPS, thus achieving the desired coverage by static sensing. PEAS, which contains two mechanisms: sensing and adaptive sleep scheduling, requires high sensor density and adjustable transmission range. PECAS is an updated version of the PEAS protocol. It posts its remaining/available hours in the response to neighbors to avoid misperceptions. The consequence is an increase in communication energy expenditure. BS [21] is a balanced-energy scheduling model designed for dense sensor networks. It distributes the sensing and communication tasks to all sensor nodes in the cluster.

We list the assumptions made by the reviewed studies in Table 1.

Table 1. The studies classification according to their assumptions.

Strategies	Network structure	Sensor placement	High density	Sensing area	Time synch.	Frequent failures	Mobility	Known location	Known distance
					Assumptions				
RIS	Flat	Grid, Uniform, Poisson	N	2-D	Y	N	N	N	N
BS	Hierarchical	Poisson	N	2-D	Y	N	N	N	Y
LDAS	Flat	Uniform	N	2-D	N	N	N	N	N
PEAS	Flat	Uniform	Y	Any	N	Y	N	N	N
PECAS	Flat	Uniform	N	Any	N	Y	N	N	N
CCP	Flat	Any	N	2-D	N	N	N	Y	N
ASCENT	Flat	Any	Y	Any	N	N	N	N	N
OGDC	Flat	Any	N	2-D	Y	N	N	Y	N
LEACH-GA	Hierarchical	Any	N	Any	Y	N	N	N	N
IBLEACH	Hierarchical	Any	N	Any	Y	N	N	N	N
ESS	Hierarchical	Any	N	Any	Y	N	N	N	N
DSS	Hierarchical	Poisson	N	2-D	Y	N	N	N	Y
IDCA	Flat	Any	N	2-D	N	N	N	Y	N

1. Y: yes; N: no
2. Time synch.: Time synchronization;

Applications differ in their necessities. Hence, the served sensor networks have diverse design objectives and priorities. We summarize these design objectives as follows.

- *Maximizing Network Lifetime*. This is a goal that is hard not to consider.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 June 2021 doi:10.20944/preprints202106.0002.v1

- *Sensing Coverage*. A network is said to achieve k-coverage if any event occurs within the jurisdiction of at least k sensors. 1-overage is generally a minimum requirement for WSN.
- *Network Connectivity*. The developers hail a model that provides particular network connectivity needed by the application. But this requires a very high sensor density.
- *Balanced Energy Consumption*. In case of a sensing coverage breach when a node runs out of power before the others, some studies endeavor to spread the energy utilization evenly among each node.
- *Simplicity*. Currently, sensors have exceptionally restricted memory space and limited computation power. Therefore, simple schemes are more popular.
- *Robustness*: Robustness measures to what extent a network can withstand downtime and crashes.

ASCENT [18], PEAS, PECAS, LEACH-GA [19], IBLEACH [20], ESS [23], OGDC [24], and BS does not regard complete coverage of the region as their primary goal. CCP [22] introduced the concept of interconnected sensor coverage. It developed an approximation algorithm to construct a topology containing a nearly optimal sensor coverage. The central controller periodically selects sensors along a path until the target area is fully covered.

Most schemes strive to achieve energy balance. For example, the Distributed self-spreading algorithm (DSS) [26] and Intelligent Deployment and Clustering Algorithm (IDCA) [27]. In the DSS, the sensor nodes are initially deployed randomly and move because of the influence exerted by nearby nodes. In the IDCA, however, the residual power of the node determines if it moves or not. The idea behind DSS and IDCA is to reduce the residual capacity differential between nodes.

Table 2. The studies classification according to the objectives

	Objectives				
Strategies	**Sensing coverage**	**Network connectivity**	**Simplicity**	**Robustness**	**Energy balance**
RIS	Full,K,asymptotic	×	×	Y	Y
BS	×	×	×	×	Y
LDAS	Partial,1,statistical	×	×	×	Y
PEAS	×	1, Hard	×	×	×
PECAS	×	1, Hard	×	×	Y
CCP	Full,K,hard	K, Hard	×	×	Y
ASCENT	×	×	Y	×	×
OGDC	Full,1,hard	1, Hard	×	Y	Y
LEACH-GA	×	×	×	Y	Y
IBLEACH	×K, Hard	×	Y	Y	
ESS	×	×	×	Y	Y
DSS	×	×	×	×	Y
IDCA	Full,original,hard	×	×	×	Y

×: not mentioned.

Many studies envisage network connectivity with sensing coverage. For example, [28–31]. When the transmission range of the sensor node is at least twice its sensing range, k-coverage leads to k-connectivity [28,29]. And typically, high connectivity ensures high robustness. But one of the results of high connectivity is that data conflicts between nodes can seriously affect data transfer rates. The results of recent research presented in [30] are not based on the assumption mentioned above that the transmission range of the sensor nodes is not less than twice their sensing range. Dhumal *et al.* [31] considered a tiered sensor network that includes sensors that could fail. Besides, they discussed the sensing coverage, network connectivity, and network diameter. In [33], the authors proposed an optimal deployment strategy to achieve a two connection that fully covers all communication and sensing ranges.

We summarize the design objectives of all studies in Table 2.

2.2. Optimization algorithms

Over the last few years, optimization algorithms have advanced considerably. New algorithms are emerging, such as the quantum tabu search algorithm [35], the artificial bee colony algorithm [36], and the adaptive mutation particle swarm optimization algorithm [37]. When addressing practical issues, we search for the global extremum of a complicated or unknown function; but just finding one local minimum of a relatively simple but very high-dimensional function can also be a formidable challenge; for example, the Multimodal Function Optimization (MFO) [38]. Heuristic algorithms, such as the particle swarm optimization (PSO) algorithm and genetic algorithm (GA), are essentially methods by trial and error. It can take many thousands or even millions of iterations, so it's not economical.

Fibonacci tree optimization algorithm (FTO)[34] is a sophisticated optimization algorithm. It resolves the optimal solution of the problem by alternating iterations of global scanning and local scanning. It fully utilizes the computer memory to save the optimization process. FTO can give a reasonable approximation of a global optimum for a function with a large search space. Furthermore, in each iteration, the golden ratio separation is used to compress the search space, so the local optimal solution can also be obtained. It particularly applies to multi-peak/multi-modal function optimization.

3. System Model

Table 3 shows the important notations used in the model.

Table 3. List of important notations in the model descriptions.

#	Description
L	Work interval
S	Sleep interval
$C(t)$	The counting process of events
λ	Poisson rate
R_{miss}	Missed event ratio
E_S	Energy expenditure for sensing
P_S	The energy expenditure of sensing in the sleep interval
P_L	The energy expenditure of sensing in the working interval
δ	The step-size increasing factor
β	The step-size diminishing factor
(X, Y)	The 2-D coordinate of a sensor
N	The number of sensors
r	The sensing radius of a sensor
A	The sensing area of a sensor

3.1. Basic assumptions

The model operates on a 2-D map with many devices, each equipped with sensors to perform specific tasks. We make the following basic assumptions:

A1: The network structure is flat. All nodes are homogeneous.
A2: The positions of sensor nodes are uniformly distributed.
A3: The sensing area is 2-D.
A4: The transmission range remains unchanged.
A5: Time is asynchronous.
A6: The sensor node is immovable.
A7: Location information is unknowable.

A8: Distance information is unknowable.

3.2. Location distribution

We assume that there are N sensors follow a uniform distribution; each has a sensing range r.

3.3. Event occurrence

We assume that the occurrence of the events satisfies a Poisson process of parameter λ. By leveraging the properties of Poisson's process, we have

$$E[C(t)] = \lambda t, \tag{1}$$

where $C(t)$ is the counting process of events. Eq. (1) means that the expected number of events occurring in the time interval t is equal to λt

The location of any event is randomly selected and thus follows a uniform distribution.

3.4. Time Synchronization

It is hard to coordinate sensors without a central controller, but otherwise the central controller incurs a performance penalty. So, sensors embrace asynchronous scheduling — every sensor autonomously decides its duty cycle without synchronizing with each other.

3.5. Design objectives

The design objective of our model is an optimization problem:

$$\arg\min_{L,S} J, \tag{2}$$

where

$$J = E[R_{miss}] + E[\bar{P}], \tag{3}$$

$E[R_{miss}]$ denotes the average event miss-rate:

$$E[R_{miss}] = E\left\{\frac{\lambda S}{\lambda S + \lambda L}\right\} = \frac{E[S]}{E[S] + E[L]}, \tag{4}$$

similarly, $E[\bar{P}]$ means the normalized average energy expenditure,

$$E[\bar{P}] = \frac{E[L] + \frac{P_S}{P_L}E[S]}{E[L] + E[S]}. \tag{5}$$

We assume that the measurement is efficient; as long as the event is in the sensor's sensing range, it is timely detected whenever an event occurs.

3.6. Scheduling model

Figure 1 shows a schematic diagram of a typical alternate sensing scheduling scheme. A sensor wakes up its sensing unit when transitioning to the working state and turns it off when the sleep interval comes. Despite the scheme's prevalence, how to adjust the intervals properly is as yet an open issue.

Now, we establish a baseline scheduling strategy: fixed-time scheduling (denoted as FT for convenience). In FT, we set the working interval and the sleep interval in advance and remain unchanged. Then, we introduce an adaptive scheduling strategy, in which

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 June 2021 doi:10.20944/preprints202106.0002.v1

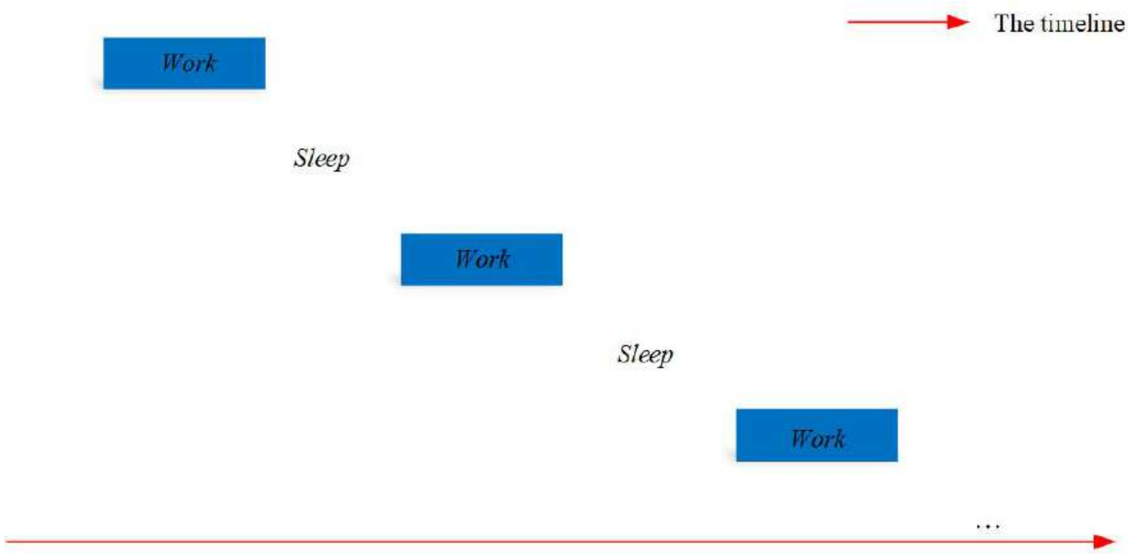


Figure 1. The diagram of alternate sensing scheduling.

we dynamically fine-tune the sleep interval as per the event statistics and user-specified parameters. The mathematical formula is

$$S(k) = \begin{cases} S_{k-1} + \delta S_k, & I_E(L_{k-1}) = 0 \\ S_{k-1} - \beta S_{k-1}, & I_E(L_{k-1}) = 1 \end{cases}, \tag{6}$$

where δ and β denotes the step-size increasing/diminishing factor, $0 \leq \delta \leq 1, 0 \leq \beta \leq 1$. $I_E(\cdot)$ is an indicator function:

$$I_E(t) = \begin{cases} 1, & \text{If an event is detected during time t} \\ 0, & \text{Otherwise} \end{cases}. \tag{7}$$

δ and β provide the ability to maintain a balance between the event capture and energy expenditure. For instance, a larger δ renders more energy-saving but also increases the probability of event missing.

3.7. Sensing coverage

Practical applications require monitoring almost the whole area of the wireless sensor network. Achieving 1-coverage is a minimum requirement (as shown in Figure 2).

Theorem 1. *Suppose there are N sensors uniformly distributed in a WSN, the sensing radius of each sensor is r, then the event miss-rate (the probability of missing an event) is given by*

$$R_{miss} = e^{-\frac{E[L]}{E[L]+E[S]}\frac{N}{A}\pi r^2}. \tag{8}$$

Proof. the probability of a random event occurred within the sensing radius of N_0 sensors is

$$P_0 = \frac{\left(\frac{N}{A}\pi r^2\right)^{N_0}}{N_0} e^{-\frac{N}{A}\pi r^2}, \tag{9}$$

The expected number of sensors within the scope of an event is

$$E[N_0] = \frac{N}{A}\pi r^2, \tag{10}$$

The expected number of sensors that are within the scope of an event and are themselves in a working state is

$$E[N_0^L] = -\frac{E[L]}{E[L]+E[S]}\frac{N}{A}\pi r^2, \tag{11}$$

Therefore, the probability of a random event being detected by a sensor is

$$1 - e^{-\frac{E[L]}{E[L]+E[S]}\frac{N}{A}\pi r^2}, \tag{12}$$

The theorem holds. □

Theorem 1 reveals that the event miss-rate is a function of the average number of sensors in each cell area and sensing scheduling.

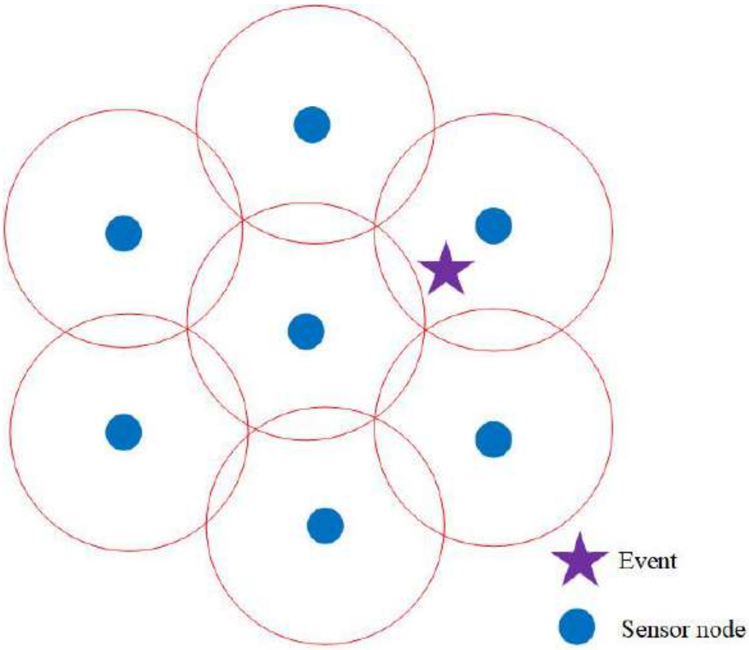


Figure 2. The diagram of 1-coverage

4. Determining Model Parameters with FTO algorithm

4.1. Fitting the objective function

Section 3.6 presents our adaptive scheduling model with two parameters to be determined: δ and β. From section 3.5, we know that the design objective of our scheduling strategy is to minimize an objective function J.

$$\arg\min_{L,S} J, \tag{13}$$

It consists of two parts, $E[R_{miss}]$ and $E[\bar{P}]$. It follows from Eq. (4) and (5) that $E[R_{miss}] = \frac{E[S]}{E[S]+E[L]}$, $E[\bar{P}] = \frac{E[L]+\frac{P_S}{P_L}E[S]}{E[L]+E[S]}$. The objective function is jointly determined by four variables: $E[S]$, $E[L]$, P_S and P_L. Because $E[S]$ and $E[L]$ are directly linked to the parameters δ and β in the model. We get

$$E[S] = \left(1 + \frac{(\delta-\beta)}{2}\right)kS_0, \tag{14}$$

$$E[L] = T_0 - E[S], \tag{15}$$

where T_0 denotes the total test duration, which is generally a fixed constant. S_0 is the initial sleep interval, which is a constant as well.

Therefore, the problem now becomes: how to establish the functional relationship between P_S,P_L and δ,β. We've been thinking about numerical fitting methods — if we have real historical sample data, we can fit out the function between P_S,P_L and δ,β.

Numerical fitting is also known as curve fitting. As for the discrete data obtained by sampling, we often want to get a continuous function (i.e., a curve) or a denser discrete equation consistent with the known data. The steps of numerical fitting are as follows:

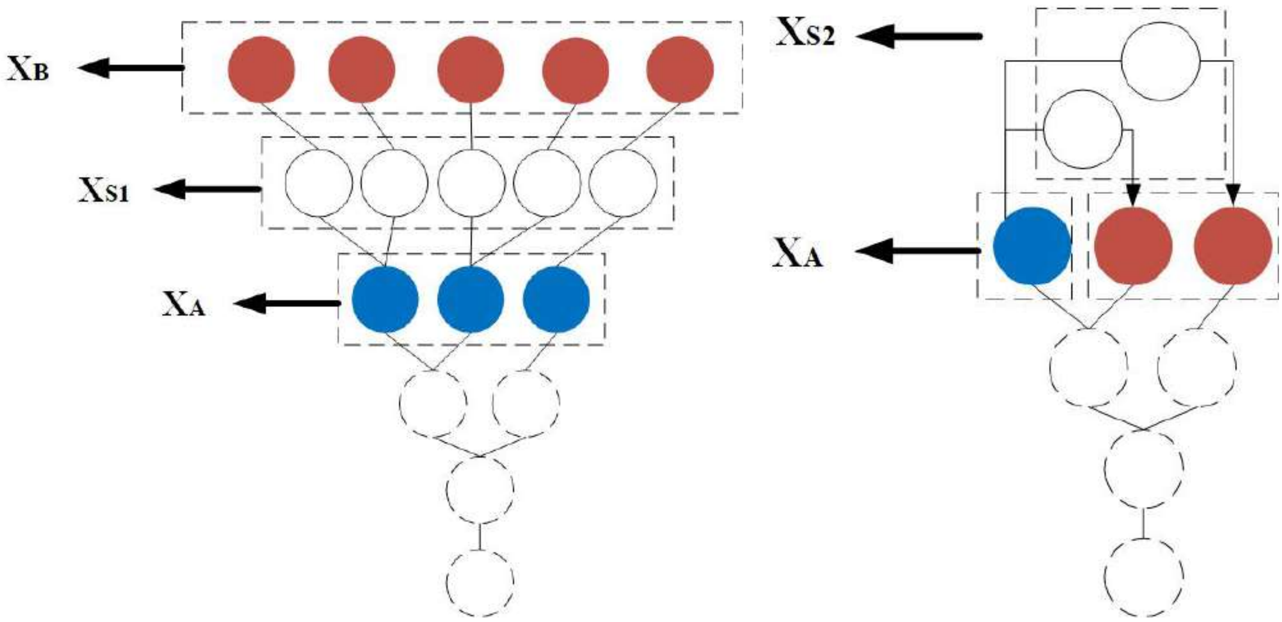
1. Presumes a functional form. Commonly used function forms include: polynomial function, exponential function, logarithmic function, trigonometric function, etc.
2. Determine the indeterminate coefficients. Using the least square method, point group center method, random fuzzy method, and so on to determine the coefficients of the given function. Of those, the least square method is the most commonly used.
3. Evaluation. The imitative effect can be measured by the mean square error (MSE) or the degree of fit (R^2).

4.2. Determination of parameters with the FTO algorithm.

The optimization problem is

$$\begin{aligned} &\arg\min_{\delta,\beta} J(\delta,\beta), \\ s.t.\quad &0 \leq \delta \leq 1, \\ &0 \leq \beta \leq 1, \end{aligned} \tag{16}$$

We know that one-dimensional Fibonacci (golden ratio) search can't proficiently take care of multi-variate issues, giving rise to Fibonacci Tree Optimization — an improved multi-dimensional search algorithm [34].



(a) global optimization stage (b) local optimization stage

Figure 3. The diagram of the branch generation process in the Fibonacci tree optimization algorithm.

Let $\mathbf{X_A}$, $\mathbf{X_B}$ and $\mathbf{X_C}$ be the vectors in D-dimensional Euclidean space. In particular, $D = 2$. $\mathbf{X_A}$ and $\mathbf{X_B}$ address the endpoints of the search component fulfilling the opti-

 doi:10.20944/preprints202106.0002.v1

mization rule, and $\mathbf{X_C}$ signifies the split points that can be resolved from searching rules. A proportion of the vectors can be determined as follows:

$$\frac{\|\mathbf{X_C}-\mathbf{X_A}\|}{\|\mathbf{X_B}-\mathbf{X_A}\|}=\frac{\|\mathbf{X_B}-\mathbf{X_C}\|}{\|\mathbf{X_C}-\mathbf{X_A}\|}=\frac{F_p}{F_{p+1}}. \tag{17}$$

where F_p denotes the pth Fibonacci number, and $\frac{F_p}{F_{p+1}}$ is equal to what's called the golden ratio.

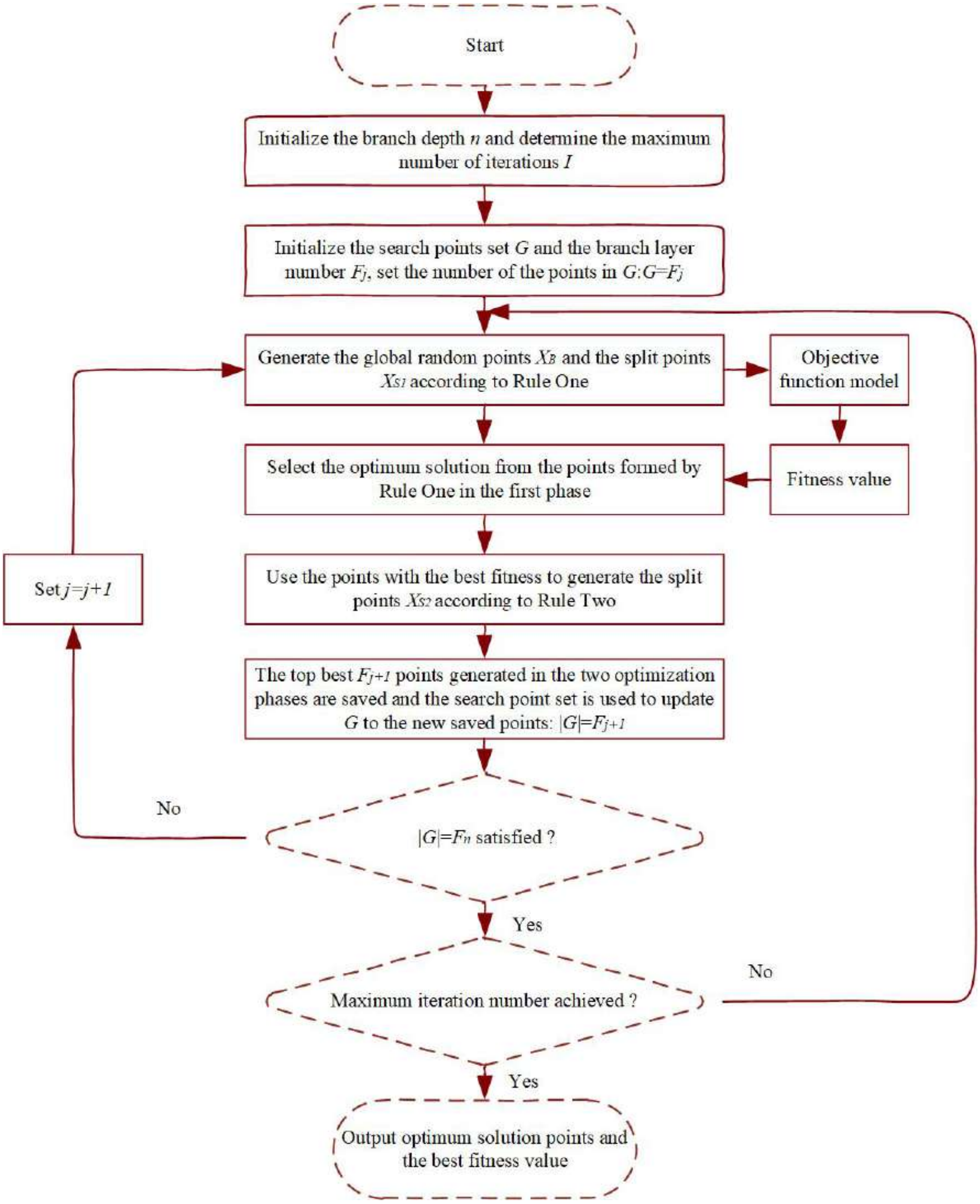


Figure 4. The flowchart of the Fibonacci tree optimization algorithm.

The fitness function determined by the endpoints in the construction ought to be assessed as

$$J(\mathbf{X_A}) < J(\mathbf{X_B}). \tag{18}$$

where $J(\cdot)$ denotes the objective function. Then, the coordinate for split point $\mathbf{X_C}$ can be calculated as

$$\mathbf{X_C} = \mathbf{X_A} + \frac{F_p}{F_{p+1}}(\mathbf{X_B} - \mathbf{X_A}). \tag{19}$$

Finding the optimal value can likewise be viewed as setting up a search component in FTO, and can be partitioned into two phases: the local optimization phase and the global optimization phase. Allow G to signify the points set of the objective function and set $|G|_{num} = F_p, i = 1, 2, \cdots, n$, where $|\cdot|_{num}$ denotes the number of points in the set, and n represents the depth of the tree. We select the point with the best fitness value. Then, in the following optimization stage, the points are rearranged by their fitness values from best to worst.

There are two search rules in FTO, which can be summed up as follows.

Rule One: Consider the endpoints $\mathbf{X_A}$ and $\mathbf{X_B}$, which are given by

$$\{\mathbf{X_A}\} = G_p = \{\mathbf{X_q} | q = [1, F_p]\} \tag{20}$$

$$\{\mathbf{X_B}\} = G_p = \left\{\mathbf{X} | \mathbf{X} \in \prod_{j=1}^{D} \left[X_{lb}^J, X_{ub}^J\right]\right\} \tag{21}$$

where G_p denotes the points set, with each point represented as $\mathbf{X_q}$, of the pth iteration, and q is the index. $\mathbf{X_A}$ absorb every point from G_p. $\mathbf{X_B}$ select F_p points from G_p randomly, where F_p denotes the population size. X_{ub}^J and X_{lb}^J are the upper and lower limits for every point. For $\forall \mathbf{X} \in \{\mathbf{X_B}\}$, $\mathbf{X}$ satisfies a uniform distribution over the span $\left[X_{lb}^J, X_{ub}^J\right]$, viz.

$$\Pr(\mathbf{X}) = U(X_{lb}, X_{ub}) = \frac{1}{X_{ub} - X_{lb}} \tag{22}$$

Using the $\mathbf{X_A}$ and $\mathbf{X_B}$ given above, the split points $\mathbf{X_{S1}}$ is solved by eq. (19).

Rule Two: Assume that $\mathbf{X_{best}}$ is the optimal point in the current iteration given by rule one, viz.

$$\mathbf{X_{best}} = BEST(G_p) \tag{23}$$

Then, let $\mathbf{X_A} = \mathbf{X_{best}}$, we have

$$J\{\mathbf{X_A}\} = \min\{J\{\mathbf{X_q}\} | q = [1, F_p]\} \tag{24}$$

$$\mathbf{X_B} = \{\mathbf{X_q} | \mathbf{X_q} \in G_p \wedge \mathbf{X_q} \neq \mathbf{X_A}\} \tag{25}$$

Hence, the split points $\mathbf{X_{S2}}$ in the local optimization stage can be determined by eq.(24) and eq.(25).

After applying the two rules above, new endpoints $\mathbf{X_A}$ and $\mathbf{X_B}$ and split points $\mathbf{X_{S1}}$ and $\mathbf{X_{S2}}$ are generated, and now we have $3F_p$ points. These points are sorted from best to worst based on the fitness value, retaining the optimal $F_p + 1$ points, while the remaining $3F_p - F_p - 1$ points are eliminated. At the end of this process, the set of search spaces for the current p iteration is updated from the remaining points and form a new set G_{p+1} for the next iteration.

Figure 3 shows the branch generation process in the Fibonacci tree optimization algorithm. The depth is initialized as expected, and the number of points in each branch layer is equal to F_p. In Figure 3, the dotted circle represents the search points set of the previous iteration; the solid red circle represents the endpoint $\mathbf{X_A}$ of the current iteration, and the solid blue circles represent the global random endpoints. Figure 3(a) shows the global optimization phase, the split points $\mathbf{X_{S1}}$ is constructed based on $\mathbf{X_B}$ and $\mathbf{X_A}$ and is

 doi:10.20944/preprints202106.0002.v1

represented by a solid white circle. Figure 3(b) shows the local optimization phase. A new split points $\mathbf{X_{S2}}$ can be obtained as per the rules.

Figure 4 shows the fundamental flowchart of the procedures for the FTO algorithm.

5. Analysis

5.1. Why FTO algorithm is chosen for parameter optimization?

First and foremost, in FTO algorithm, the growth of Fibonacci tree is based on the optimal nodes of last generation, which is actually a process of competitive elimination. If there are peaks with different heights, the smaller peaks will be eliminated in the optimization process. Therefore, the global optimal value can be achieved.

More importantly, in each iteration, the golden ratio separation is used to compress the search space, so the local optimal solution can also be obtained. It is particularly suitable in optimizing multi-peak/multi-modal functions.

Last but not the least, it fully utilizes the computer memory to save the optimization process and thus is traceable.

5.2. Why we claim FTOS strategy is superior than other scheduling strategies?

Some of the latest scheduling strategies, such as LEACH, work under strict assumptions; for example, the network structure is flat, nodes are densely distributed, time is synchronized, and so forth.

The FTOS strategy does not require time synchronization (centralized scheduling); each node independently determines the duty cycle; moreover, node density is not important and the location and distance information is not needed.

In addition, the FTOS strategy determines the scheduling parameters as per the occurrence of events in the previous period, which is highly adaptable.

Once the parameters are determined, they remain unchanged in the coming hours, you can easily deploy them.

6. Experiment

6.1. Experimental setup

We set up a wireless sensor network comprising 3 nodes ($N = 3$) covering a 5m×5m lawn area (see Figure 5). Each node owns a single-chip microcomputer with a STM32F103ZET6 processor and a diffuse reflection photoelectric infrared proximity sensor. The technical parameters of the sensor are presented in Table 4.

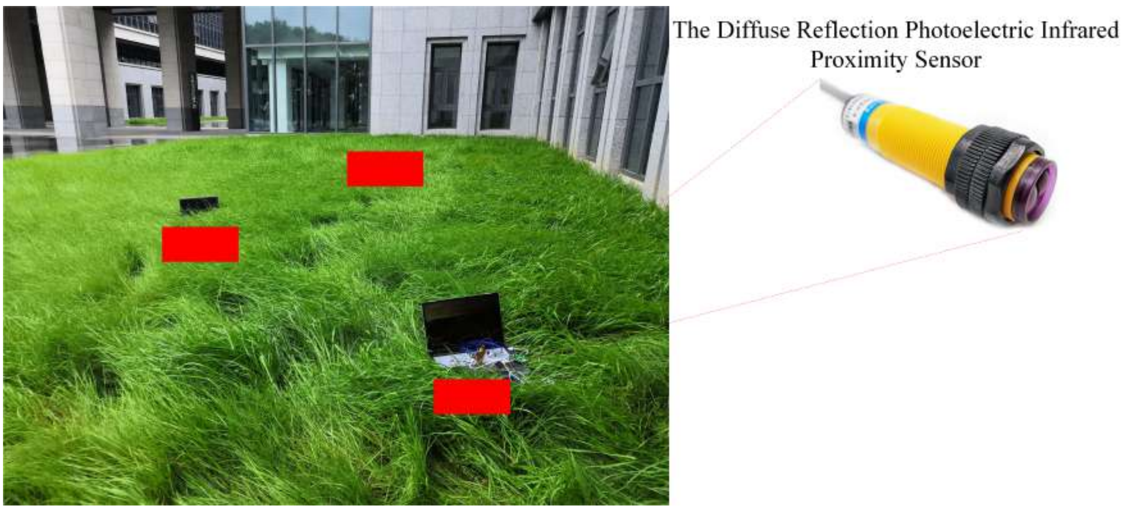


Figure 5. The The experiment scene.

We randomly throw objects in the area to simulate the event. The diffuse reflection photoelectric infrared proximity sensor that has a sensing range of $r = 1$m opens the

 doi:10.20944/preprints202106.0002.v1

Table 4. The technical parameters for the diffuse reflection photoelectric infrared proximity sensor.

Parameter	Value
Product model	E3F-DS100P1
Operating voltage	DC 6~36 V
Operating current	200 mA
Response frequency	50 Hz
Sensing object	Any opaque object
Sensing radius	10~100 cm, adjustable

switch when there is an object to block; the processor then issues a command to pull the buzzer.

Since the working voltage of the single-chip microcomputer is fixed to 5V, the power formula is $P = UI$, so that the current reflects the size of the power consumption. Consequently, the energy is measured by the reading of the ammeter placed on each microcontroller.

6.2. Train

6.2.1. Objective function fitting

We assume that the sensor node determines its sleep interval based on the adaptive scheduling model, where the parameters δ and β are randomly generated on $[0, 1]$. The polynomial function, double-exponential function, and checkmark function are respectively used to fit the data(as shown in Table 5), and the determination of the coefficients is done by the nonlinear least square method.

Table 5. The fitting functions.

Function	Form
Polynomial function	$ax^5 + bx^4 + cx^3 + dx^2 + ex + f$
Double-exponential function	$ae^{bx} + ce^{dx}$
Checkmark function	$ax + \frac{b}{x} + c$

Table 6. The degree of fit (R^2).

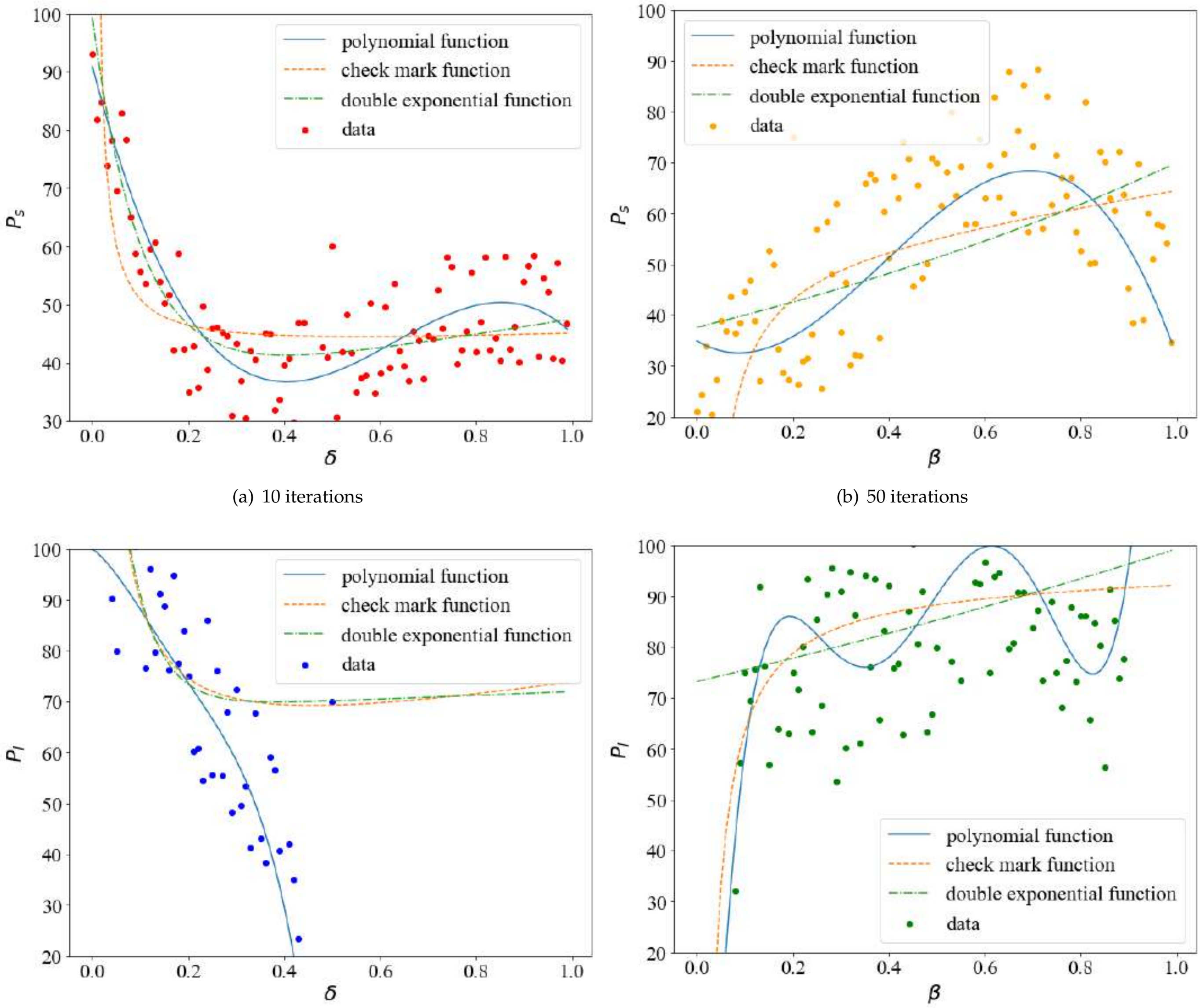
Cases	Polynomial function	Double-exponential function	Checkmark function
P_S and δ	0.73	**0.76**	0.70
P_S and β	**0.63**	0.39	0.32
P_L and δ	**0.91**	0.01	0.004
P_L and β	**0.85**	0.61	0.02

We generated 100 samples, recorded their average energy expenditure during sleep intervals, P_S, and during working intervals, P_L, and corresponding parameters δ and β . Figure 6 shows the fitting curves. Table 6 enlists the degree of fit, R^2.

Now that we have determined the functional relations between P_S, P_L and δ, β, the functional expression of the objective function J can be deduced through the following relationships:

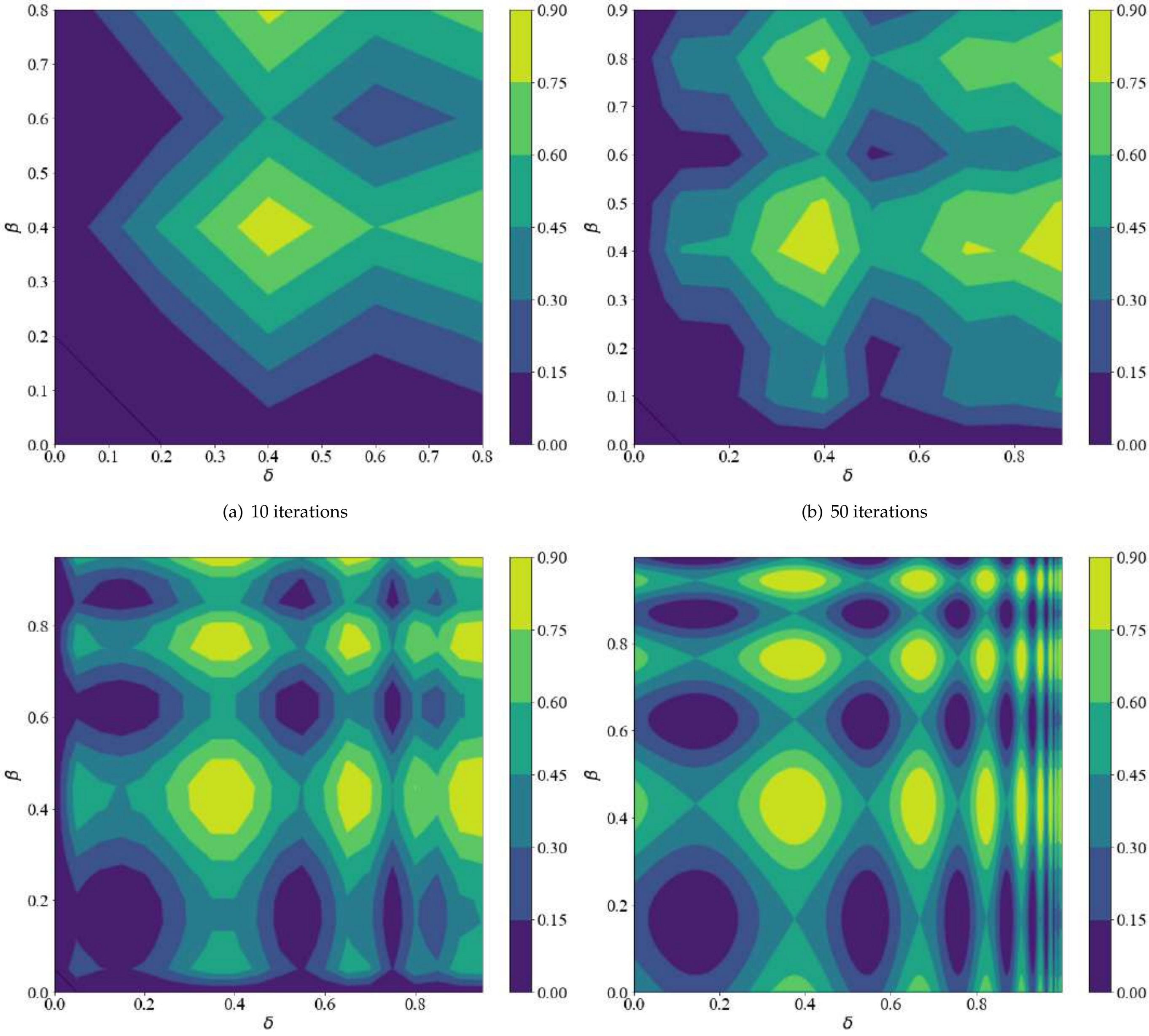
$$
\begin{cases}
J = E[R_{miss}] + E[\bar{P}] \\
E[R_{miss}] = \dfrac{E[S]}{E[S] + E[L]} \\
E[S] = \left(1 + \dfrac{\delta - \beta}{2}\right) k S_0 \\
T_0 = E[S] + E[L] \\
E[\bar{P}] = \dfrac{E[L] + \frac{P_S}{P_L} E[S]}{E[S] + E[L]} \\
P_S = f_1(\delta) \\
P_S = f_2(\beta) \\
P_L = f_3(\delta) \\
P_L = f_4(\beta)
\end{cases}
\tag{26}
$$

where S_0 and T_0 are constants, and $f_1 \sim f_4$ are fitted functions.



(a) 10 iterations (b) 50 iterations (c) 200 iterations (d) 500 iterations

Figure 6. The fitting curves of P_S, P_L and δ, β.



(a) 10 iterations

(b) 50 iterations

(c) 200 iterations

(d) 500 iterations

Figure 7. The contour plots for different iterations.

6.2.2. Optimization of the step-size parameters

Observing the fitting curves given above, we realize that this is a bivariate multi-peak function, and it is appropriate using the FTO algorithm.

Figure 7 shows the optimization process of the FTO algorithm, and the values of the horizontal and vertical axis represent the values of δ and β respectively. The darker the color of the contour map is, the smaller the value of objective function J, which is the area to be determined. The algorithm stops after 500 iterations, and Figure 8 provides the final comparison of the peak areas. We can see that the peak area in the lower-left corner is the largest, and is where the optimal point is most likely to appear. The central point of the aforementioned area was taken as the ultimate global optimization point with the coordinate of $(\delta, \beta) = (0.15, 0.15)$.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 June 2021 doi:10.20944/preprints202106.0002.v1

As a comparison, we also implemented Particle Swarm Optimization (PSO) algorithm, Genetic Algorithm (GA), Comprehensive Learning PSO (CLPSO) algorithm, Differential Evolution (DE) algorithm, and Artificial Bee Colony (ABC) algorithm. Table 7 outlines the parameters for each algorithm, and Table 8 gives the comparison of the optimization results. We can see that the FTO algorithm outperforms others in most iteration rounds.

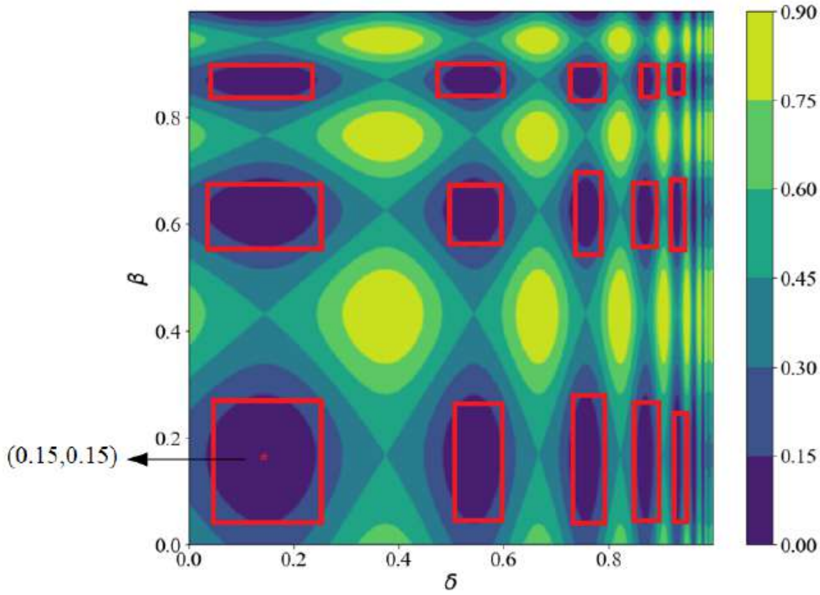


Figure 8. The determination of the most prominent peak area and global optimization point

Table 7. The algorithm parameters adopted.

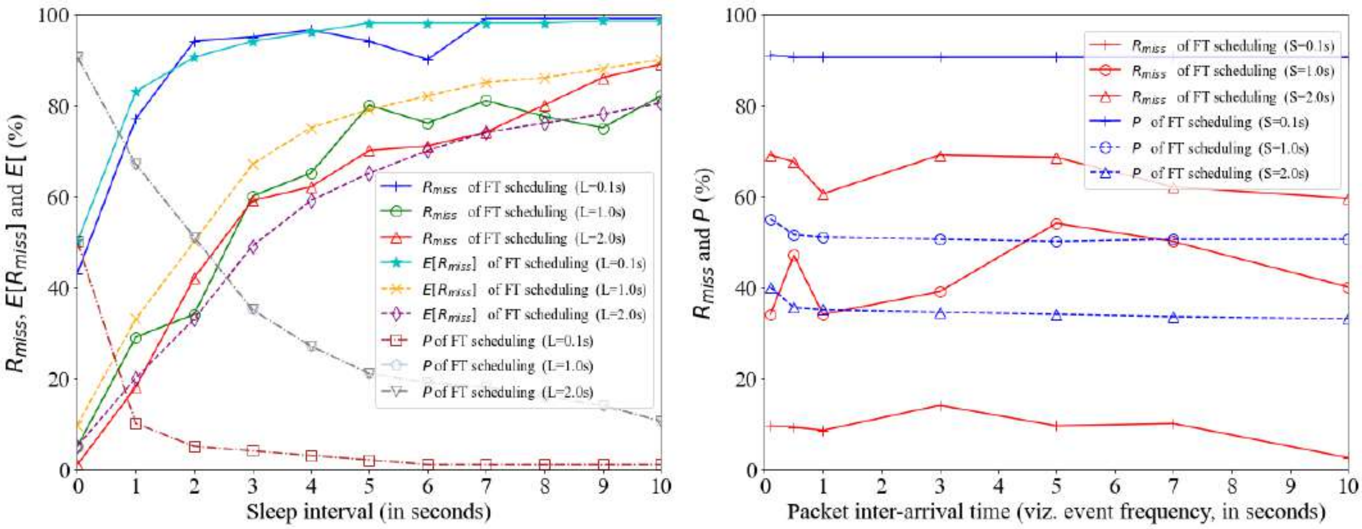
Algorithm	Parameter	Value
PSO	Population size	20
	Cognitive ratio	2
	Social coefficient	2
	Inertia weight	0.4~0.9
GA	Population size 20	
	Mutation probability	0.05
	Cross probability	0.7
	Rate of chromosome elite	0.2
CLPSO	Learning probability	0.05~0.5
	Population size	20
	Cognitive ratio	2
	Social coefficient	2
	Inertia weight	0.4~0.9
DE	Population size	20
	Scaling factor	0.6
	Crossover rate	0.8
ABC	Colony size	20
	Onlooker bees percentage	50%
	Scout bees	1
FTO	Nested branch depth	2
	Total branch depth	6
	Search space	[0,1]
	Max iterations	1000
	Precision	0.001

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 June 2021 doi:10.20944/preprints202106.0002.v1

Table 8. The optimization results.

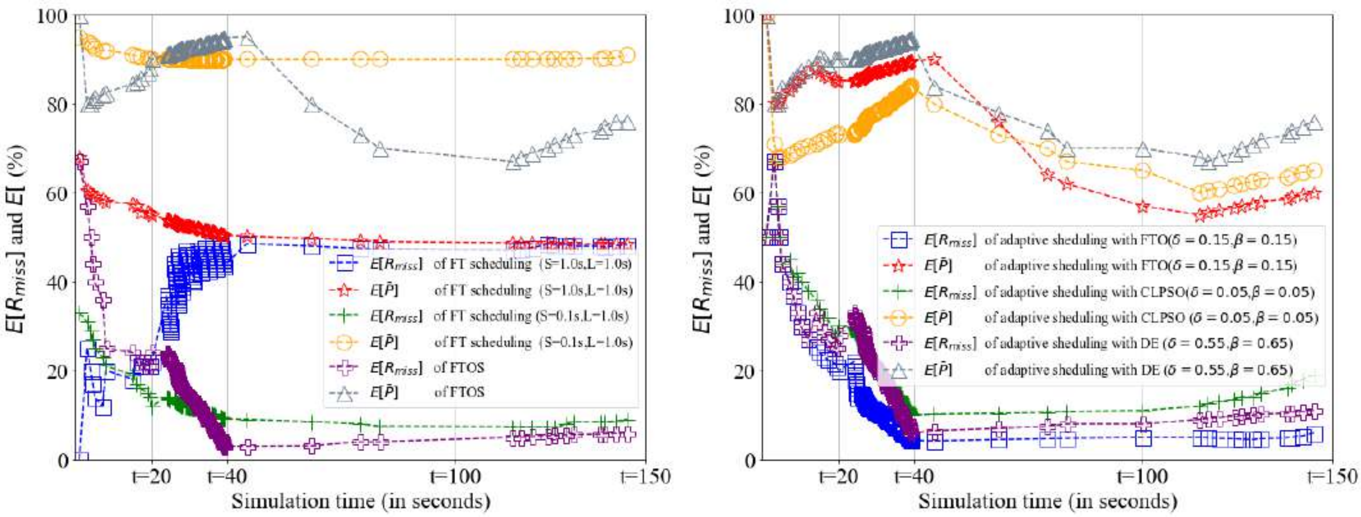
Algorithm	iter=10			iter=50			iter=200			iter=500		
	min J	δ	β	min J	δ	β	min J	δ	β	min J	δ	β
ABC	0.73	0.51	0.52	0.43	0.59	0.3	0.163	0.56	0.32	0.137	0.19	0
DE	0.78	0.51	0.1	0.55	0.18	0.55	0.174	0.36	0.9	0.067	0.55	0.65
GA	0.81	0.49	0.2	0.29	0.21	0.11	0.167	0.83	0.44	0.142	0.15	0
PSO	0.57	0.45	0.49	0.61	0.29	0.38	0.170	0.95	0.84	0.169	0.19	0.6
CLPSO	0.61	0.32	0.1	0.18	0.18	0.74	0.173	0.28	0.9	0.129	0.05	0.05
FTO	0.54	0.47	0.1	0.43	0.17	0.86	0.108	0.48	0.1	0.060	0.15	0.15

6.3. Test



(a) (b)

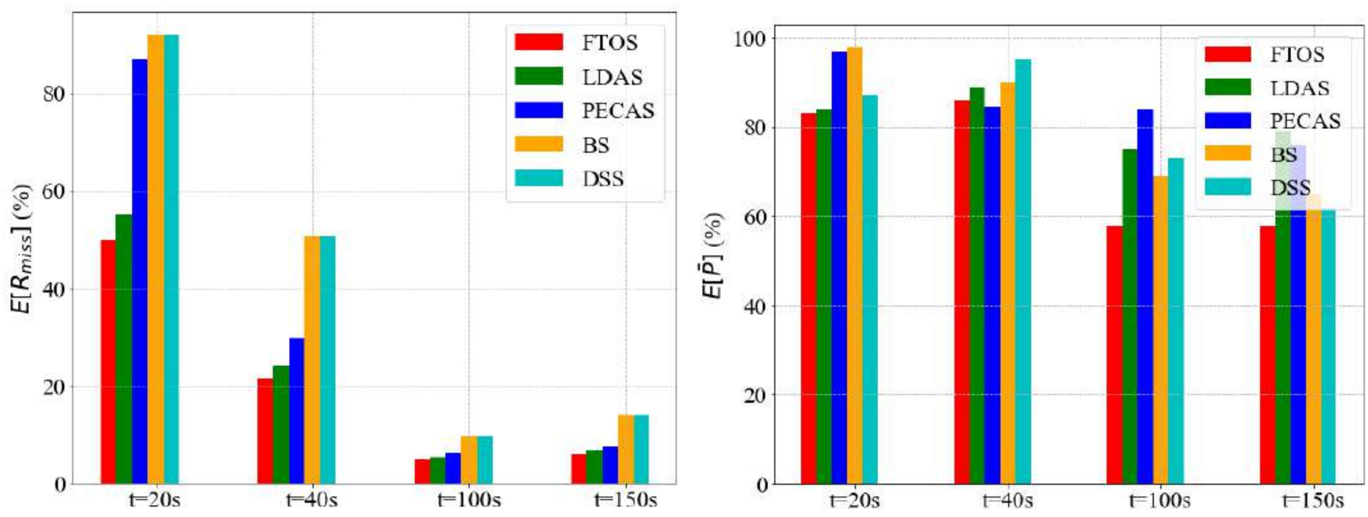
Figure 9. The event miss-rate and energy expenditure against sleep interval and event frequency in the FT scheduling. (a) against sleep interval (with a fixed Poisson rate of events and working interval); (b) against packet inter-arrival time (with a fixed working interval).



(a) (b)

Figure 10. The expected event miss-rate and normalized average energy expenditure. (a) FT scheduling strategy with two sets of parameters ($S = 1, L = 1$, and $S = 0.1, L = 1$) and the proposed FTOS strategy ($\delta = 0.1, \beta = 0.5$); (b)FTOS strategy with parameters generated by different optimization algorithms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 1 June 2021 doi:10.20944/preprints202106.0002.v1



(a) $E(\bar{R_{miss}})$ (b) $E(\bar{P})$

Figure 11. The expected event miss-rate and the normalized average energy expenditure of different sensing scheduling strategies at different time scales.

Once the parameters of the strategy are determined, they remain unchanged in the coming hours. We test the performance in terms of the event misses, energy expenditure and sensing coverage.

6.3.1. Event misses and energy expenditure

In this experiment, the generated events follow a Poisson process with Poisson rate λ that changes over time. In the first 20 seconds, $\lambda = 1$, then increases to 10 in 20~40s, then decreases to 0.1 in 40~100s, and then back to 1 after that. The purpose of varying the Poisson rate is to establish a real environment.

Figure 9(a) shows the event miss-rate and energy expenditure for different sleep intervals with Poisson rate $\lambda = 1$. A considerable sleep interval accomplishes better energy utilization, yet it increases the probability of missing events. However, Figure 9(a) also implies that the event miss behavior and energy expenditure are not influenced much when the sleep interval is long enough. Figure 9(b) shows that the event miss-rate and energy expenditure are less likely to be affected by packet inter-arrival time (i.e., event frequency). From the analysis above, we can safely conclude that it is the sensing scheduling strategy that mainly determines the event miss-rate and energy expenditure in FT mode.

Figure 10(a) shows the expected event miss-rate and normalized average energy expenditure of the proposed FTOS and FT strategies with two sets of parameters $S = 1, L = 1$, and $S = 0.1, L = 1$. We found that the FT strategy with improperly setting ($S = 1, L = 1$) performed unsatisfactorily, while the one with a better setting ($S = 0.1, L = 1$) yielded much better results. However, selecting the appropriate parameters in the FT strategy requires prior knowledge of the course of events, whereas that in the FTOS strategy does not. More importantly, the proposed FTOS strategy performs better than the FT strategy for both the expected event miss-rate and normalized average energy expenditure.

Figure 10(b) shows the expected event miss-rate and normalized average energy expenditure for FTOS strategy with parameter settings optimized by different optimization algorithms including FTO ($\delta = 0.15, \beta = 0.15$), DE ($\delta = 0.55, \beta = 0.65$) and CLPSO ($\delta = 0.05, \beta = 0.05$). Despite this, the proposed FTOS strategy works well over a wide

 doi:10.20944/preprints202106.0002.v1

range of parameter settings and demonstrates some adaptivity; however, it continues to differ in performance. All settings generated by FTO are the best.

For comparison, we've implemented other sensing scheduling strategies as well. Figure 11 shows the results. We find that FTOS has the lowest event miss-rate among all sensing scheduling strategies and consumes the bare minimum of energy. Fortunately, however, over time (t=20~150s), all sensing scheduling strategies worked, i.e., both the event miss-rate and average energy expenditure declined to varying degrees.

6.4. Sensing coverage

We assume that all sensor nodes have the same energy budget, and that the measuring criteria of sensing coverage are the event miss-rates. The parameters of the FT strategy are $S = 0.1$ and $L = 1$, and the parameters of the proposed FTOS strategy are $\delta = 0.15$ and $\beta = 0.15$. The results are summarized in Figure 12(a)(different N) and Figure 12(b) (different r). FTOS has a lower event miss rate than FT on all experiments. Figure 12 (a) shows that as N increases, the difference between FT and FTOS gradually decreases due to the rise in node density. Figure 12(b) indicates a slight difference between FT and FTOS if the sensing radius r is very small. As r increases, the sensing scheduling strategy begins to serve as an efficient instrument to expand coverage, and the difference between FT and FTOS starts to increase.

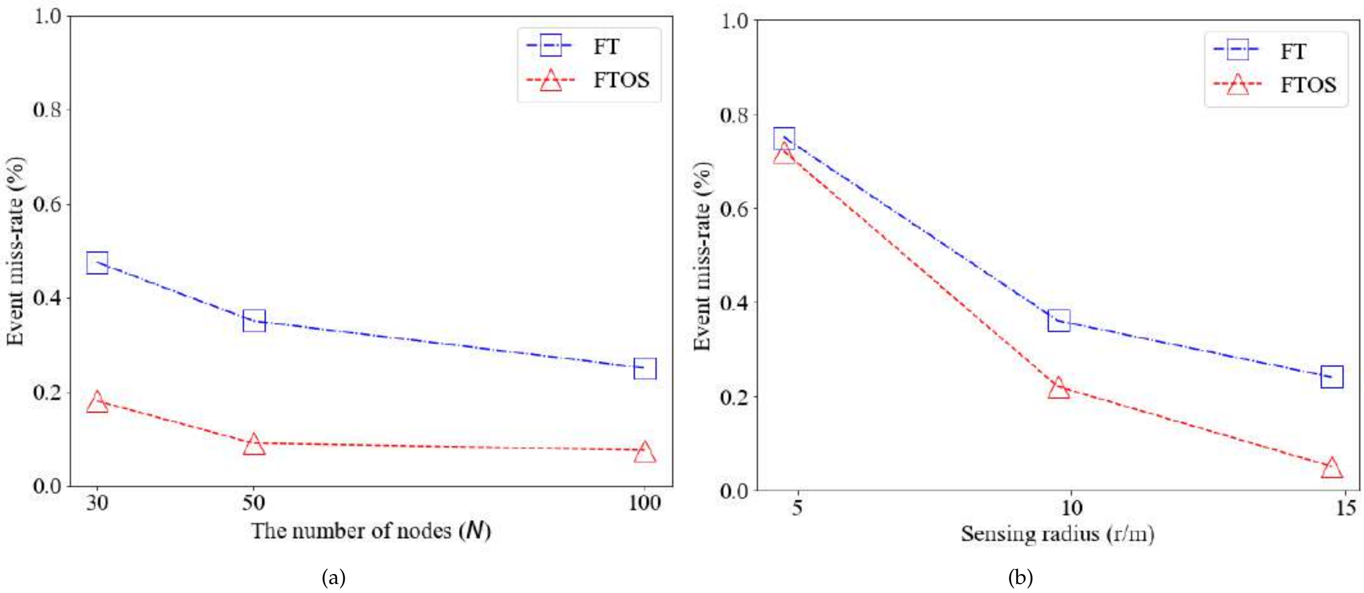


Figure 12. The event miss-rate against the number of sensor nodes and sensing radius. (a) The relationship between the event miss-rate and the number of sensor nodes (r is fixed and $r = 15m$); (b) The relationship between the event miss-rate and sensing radius (N is fixed and $N = 100$).

7. Deficiencies and Unresolved Issues

The research of this paper is still flawed in the following aspects:

1. Selection of fitting functions. Some simple fitting functions are selected by experience to simulate the relationship between variables related to the objective function and the parameters of the scheduling strategy. It is true that the results are relatively good, but other fitting functions are still worth trying.
2. Selection of sensors. The diffuse reflection photoelectric infrared proximity sensor, E3F-DS100P1, has a sensing radius of about 1 meter. If the budget is heavy enough, we suggest using sensors with a larger sensing radius and more accuracy, e.g., laser radar.

3. Too few nodes. Because of the insufficient number of devices we have, we've only configured a 3-node network. The experiments with more nodes would be more convincing.

Furthermore, the dynamics of network traffic were not analysed. And, it is interesting to associate our strategy with other approaches, such as the follow-up of specific targets in [33], which is also a field of future efforts.

8. Conclusion

The FTOS strategy proposed in this paper considers the trade-off between energy availability and fault tolerance of event monitoring. The experimental results show that the FTOS strategy can achieve the scheduling goal of low energy expenditure and low event miss rate. It performs better than LDAS, BS, DSS, and PECAS, which are some scheduling strategies of the same design objectives. The experimental results also show that the step-size parameters optimized by the FTO algorithm are better than those optimized by PSO, GA, etc. We also guide the actual deployment. It should be noted that when applied in practice, the FTOS strategy can be generalized only if the step-size can be systematically formulated.

Funding: This research was funded by the National Natural Science Foundation of China under Grant numbers 61832014.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Research data are be obatained by sending an email to wuliangshun@whu.edu.cn

Acknowledgments: The authors gratefully acknowledge Dr. Duan for providing source codes of some optimization algorithms.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Phan, K. T.; Huynh, P.; Nguyen, D. N.; Ngo, D. T.; Hong, Y.; Le-Ngoc, T. Energy-Efficient Dual-Hop Internet of Things Communications Network With Delay-Outage Constraints. *IEEE Transactions on Industrial Informatics*, **2021**, *17*, 4892-4903.
2. Boddu, N.; Boba, V; Vatambeti, R. A Novel Georouting Potency based Optimum Spider Monkey Approach for Avoiding Congestion in Energy Efficient Mobile Ad-hoc Network. *Wireless Pers Commun* **2021**.
3. Sheng, Z.; Mahapatra, C.; Zhu, C.; Leung, V. C. M. Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT. *IEEE Access* **2015**, *3*, 622-637.
4. Solis, F; Fernández Bocco, Á; Galetto, AC; Passetti, L; Hueda, MR; Reyes, BT. A 4GS/s 8-bit time-interleaved SAR ADC with an energy-efficient architecture in 130 nm CMOS. *Int J Circ Theor Appl* **2021**, 1-15.
5. Li, H.; Wang, Z.; Wang, H.An energy-efficient power allocation scheme for Massive MIMO systems with imperfect CSI. *Digital Signal Processing* **2021**, *112*, 1051-2004.
6. Begum, S.; Wang, S.; Krishnamachari, B.; Helmy, A. ELECTION: Energy-efficient and low-latency scheduling technique for wireless sensor networks. In Proceedings of the 29th Annual IEEE Conference on Local Computer Networks (LCN). Tampa, FL, Nov. 2004.
7. Dantu, R.; Abbas, K.; ONeill II, M.; Mikler, A. Data centric modelling of environmental sensor networks. In Proceedings of IEEE Globecom. Dallas, TX, 2004; 447-452.
8. Lee, J.; Lee, D.; Kim, J.; Cho, W.; Pajak, J. A dynamic sensing cycle decision scheme for energy efficiency and data reliability in wireless sensor networks. In Lecture Notes in Computer Science, 2007; 4681, 218-229.
9. Jain, A.; Chang, E. Y. Adaptive sampling for sensor networks. In Proceedings of the International Workshop on Data Management for Sensor Networks, Toronto, Canada, 2004; 72, 10-16.
10. Jothiraj, S.; Balu, S.; Rangaraj, N. An efficient adaptive threshold-based dragonfly optimization model for cooperative spectrum sensing in cognitive radio networks. *International Journal of Comunication Systems* **2021**, 1-11.
11. Chen, S.; Shao, D.; Shu, X.; Zhang, C.; Wang, J. FCC-Net: A Full-Coverage Collaborative Network for Weakly Supervised Remote Sensing Object Detection. *Electronics* **2020**, *9*, 1356.
12. Franceschi, S.; Chirici, G.; Fattorini, L.; Giannetti, F.; Corona, P. Model-assisted estimation of forest attributes exploiting remote sensing information to handle spatial under-coverage. *Spatial Statistics* **2021**,*41*.
13. He, T. Energy-efficient surveillance system using wireless sensor networks. In MobiSys 04, Boston, MA, 2004.

14. Wu, K. Lightweight deployment-aware scheduling for wireless sensor networks. *ACM/Kluwer Mobile Networks and Applications (MONET)* **2005**, *10*, 837-852.
15. Zhang, Y.; Zhang, D.; Vance, N.; Li, Q.; Wang, D. A Light-weight and Quality-aware Online Adaptive Sampling Approach for Streaming Social Sensing in Cloud Computing. In Proceedings of the 24th IEEE International Conference on Parallel and Distributed Systems (ICPADS), Singapore, SINGAPORE; Dec 11-13, 2018.
16. Ye, F.; Zhong, G. S.; Zhang, L. PEAS: A robust energy conserving protocol for long-lived sensor networks. In Proc. IEEE Int. Conf. Network Protocols (ICNP), Paris, France, 2002.
17. Liu, C. H.; Zhao, J.; Zhang, H.; Guo, S.; Leung, K. K.; Crowcroft, J. Energy-Efficient Event Detection by Participatory Sensing Under Budget Constraints. *IEEE Systems Journal* **2017**, *11*, 2490-2501.
18. Friderikos, V.; Papadaki, K.; Wisely, D.; Aghvami, H.A. Non-Independent Randomized Rounding for Link Scheduling in Wireless Mesh Networks. In the Proceedings of the 64th IEEE Vehicular Technology Conference, Montreal, CANADA; Sep 25-28, 2006.
19. Liu, J.L.; Ravishankar, C.V. LEACH-GA: Genetic Algorithm-based energy efficient adaptive clustering protocol for Wireless Sensor Networks. *International Journal of Machine Learning and Computing* **2011**, *1*, 79-85.
20. Salim, A.; Osamy, W.; Khedr, A.M. IBLEACH: Intra-balanced Leach protocol for Wireless Sensor Networks. *Wireless Network* **2014**, *20*, 1515-1525.
21. Hanef, M.; Deng, Z. Design challenges and comparative analysis of cluster based routing protocols used in Wireless Sensor Networks for improving network lifetime.*Adv Inf Sci Serv Sci* **2012**, *4*, 450-459.
22. Alam, K. M.; Kamruzzaman, J.; Karmakar, G.; Murshed, M. Dynamic adjustment of sensing range for event coverage in wireless sensor networks. *Journal of Network and Computer Applications* **2014**, *46*, 139-153.
23. Mishra, S; Yaduvanshi, R; Dubey, K; Rajpoot, P. ESS-IBAA: Efficient, short, and secure ID-based authentication algorithm for wireless sensor network. *Int J Commun Syst.* **2021**, *34*.
24. Wang, G.; Cao, G.; La Porta, T. Movement-assisted sensor deployment. IEEE INFOCOM, Hong Kong, China, 2004.
25. Kim, Y; Park, E.; Song, Y.; Han, Y. Energy-efficient sensing mechanism for licensed-assisted access under non-saturated traffic condition. In Proceedings of the 23rd Asia-Pacific Conference on Communications (APCC), 2017.
26. Heo, N.; Varshney, P. K. A distributed self spreading algorithm for mobile wireless sensor networks. In IEEE Wireless Communications and Networking Conference, New Orleans, LA; Mar 16-20, 2003.
27. Song, C. Selective CS: An Energy-Efficient Sensing Architecture for Wireless Implantable Neural Decoding. *IEEE Journal on Emerging & Selected Topics in Circuits & Systems* **2018**, 1-1.
28. Wang, C.; Song, T.; Wu, J.; Jiang, W.; Hu, J. Energy-Efficient Optimal Sensing and Resource Allocation of Soft Cooperative Spectrum Sensing in CRNs. In proceedings of the 11th International Conference on Wireless Communications and Signal Processing (WCSP), 2019; 1-6.
29. Shakkottai, S.; Srikant, R.; Shroff, N. Unreliable sensor grids: Coverage, connectivity, and diameter. In Proceedings of IEEE INFOCOM, San Francisco, CA, 2003.
30. Rebiha, S.; Fouzi, S. Energy-efficient coverage and connectivity of wireless sensor network in the framework of hybrid sensor and vehicular network. *International Journal of Computers and Applications* **2020**, 1-11.
31. Dhumal, S.; Shetty, B.S. Energy-Efficient Coverage and Sensor Localization for Scheduling. **2019**, 537-541.
32. Zhao, Q.; Zhu, Y.; Zhu, H.; Cao, J.; Xue, G.; Li, B. Fair energy-efficient sensing task allocation in participatory sensing with smartphones. In Proceedings of IEEE Conference on Computer Communications, 2014; 1366-1374.
33. Sun, M. Adaptive Sensing Schedule for Dynamic Spectrum Sharing in Time-Varying Channel. *IEEE Transactions on Vehicular Technology* **2018**, *67*, 5520-5524.
34. Zhang, H.; Zeng, F. A Fibonacci Branch Search (FBS)-Based Optimization Algorithm for Enhanced Nulling Level Control Adaptive Beamforming Techinique. *IEEE Access* **2019**, *7*, 160800-160818.
35. Chen, X.; Jiang, H. Research of quantum tabu search algorithm. *Acta Electronica Sinica* **2013**, *41*, 2161-2166.
36. Zhu, B.; Zhu, F. Discrete artificial bee colony algorithm based on logic operation. *Acta Electronica Sinica* **2015**, *43*, 2161-2166.
37. Lu, Z.; Hou, Z. Particle swarm optimization with adaptive mutation. *Acta Electronica Sinica* **2004**, *32*, 416-420.
38. DAS, S.; MAITY, S. Real-parameter evolutionary multimodal optimization-A survey of the state of the art. *Swarm and Evolutionary Computation* **2011**, *2*, 71-78.