Sensor Network Scheduling for Energy Efficiency using a Simulation-based Genetic Algorithm with Variable-Length Chromosome

Van-Phuong Ha[1,2], Trung-Kien Dao[1,*], Ngoc-Yen Pham[1], and Minh-Hoang Le[1]

A R T I C L E I N F O

Article history:
Received: 5 March 2021
Revised: 8 June 2021
Accepted: 8 July 2021

Keywords:
Wireless sensor networks
Variable-length chromosome
genetic algorithms
Schedule optimization
Planning optimization
Energy efficiency

A B S T R A C T

In monitoring applications using sensor networks, sensor network optimization is very important and has received much attention, as an effort to lowering the power consumption and maximizing the network lifetime. However, this problem is usually too complex to be solved by deterministic methods. On the other hand, metaheuristic search strategies such as genetic algorithms (GAs) can find the optimal solution efficiently with a stochastic approach. The most important advantage of GAs is its robustness to discrete and noisy fitness functions. However, for a time-based sensor network, the schedule cannot be represented by a fixed-length parameter vector that is required by traditional genetic algorithms. Therefore, in this paper, a technique using variable-length chromosome (VLC) is introduced to address this problem. In addition, to apply metaheuristic algorithms, it is also necessary to have a sensor network simulation platform that supports the estimation of energy-related activities. Simulation results show that, with help of carefully defined fitness functions, the proposed scheme can evolve the individuals in the population effectively and consistently from generation to generation toward optimal ones.

1. INTRODUCTION

Nowadays, practical applications in many fields require the application of wireless sensor networks such as environmental monitoring, healthcare, industrial production inspection, agriculture, energy, transportation, security, military as well as in civil applications [1]. Thanks to the advantages such as flexibility, customizability, easy deployment in large-scale and complex environments, wireless sensor networks have been becoming more and more widely used.

In research, development, and application of sensor networks, optimization has an important role, where the problems are usually related to energy consumption, network coverage, communication routing, network lifetime, etc. The optimization objectives are also very diverse. For example, in maximizing network lifetime, the criteria can be defined differently, either the network is considered to exist when only a few nodes remain active or when its perceptible range is above a certain threshold [2]. Moreover, a network can be heterogeneous in terms of sensor type, the number of nodes is large, the deployment environment is broad and complicated. Therefore, sensor network optimization problems are generally complex as they usually involve multiple subjects.

Among network optimization problems, schedule optimization is commonly used in deploying sensor networks with goal to reduce the energy consumption, increase the network lifetime… while guarantee its activities to satisfy the constraints specified by applications. Basically, this is achieved based on timely programming the working modes in each node individually so that they cooperate and accomplish intended tasks. However, while all strategies have a common design objective to maximize network lifetime, the mechanisms to solve this problem are as diverse as the techniques used in sensor network deployment due to different assumptions considered in the context of different applications [3]. Therefore, the mechanisms can be categorized in many ways: static and dynamic, centralized and distributed, communication-based and communication-less cooperation, hierarchical and non-hierarchical, etc.

As an example of communication-based cooperation, Miller and Vaidya in [4] developed a special MAC-layer power saving scheme using a wake-up radio to minimize the transceivers' idle time, hence reduce the communication power consumption in a non-hierarchical network. A two-radio architecture is used so that a sensor can wake-up a neighbor with a trigger and send its packets to that destination. In another study [5], Nawaz et al. developed a cooperative scheduling mechanism with

[1]MICA Institute (HUST – Grenoble INP), Hanoi University of Science and Technology, Hanoi, Vietnam.
[2]Faculty of Electrical Engineering Technology, Hanoi University of Industry, Hanoi 100000, Vietnam.
*Corresponding author: Trung-Kien Dao; E-mail: trung-kien.dao@mica.edu.vn.

physical-layer communication scheme to avoid collisions of multiple data flows in large-scale networks using OFDM (orthogonal frequency-division multiplexing) to reduce the energy consumption. In [6], Nguyen et al. applied sleeping schedules for sensor nodes together with cognitive radio techniques to reduce frequency band usage, and compressed sensing to reduce data transmission, hence cut off energy consumption in the network. The cooperative approach is generally more flexible and responses more adaptively in complex applications when external factors change quickly and working context is highly unpredictable. However, in typical application classes such as surveillance, monitoring, where these conditions are more deterministic, the transmission overhead for cooperation messages would unnecessarily consume energy and shorten the nodes' lifetime.

Therefore, whenever possible, communication-less cooperative planning would be more optimal in terms of energy effectiveness. In principle, this can be fulfilled with the help of strategies that are based on information coming from ambient factors [7], [8] where time is one of the mostly used for its availability and the simplicity in implementation. In this study, a time-based schedule optimization problem is considered, where sensor nodes are of active-then-sleep type. As time schedule is discrete by nature, searching for the optimal solution using analytical methods is unsuitable. We aim to approach this problem using genetic algorithms (GAs) [9] as a generic searching mechanism for global optimal solution. However, as the network schedule is flexible and cannot fit into classical GAs because their chromosomes are fixed in length, a modified GA with variable-length chromosome (VLC) extension is developed to address this problem, and is introduced in this paper.

Several variations of VLC-GAs have been introduced in the literature for several applications. In [10], the authors proposed a VLC to represent QoS-aware multi-path web-service composition plans so that the optimal path can be determined using a GA. To do this, the customer service requirements are modeled using a directed graph, then the service selection and composition problem is transformed into a pattern matching problem of service travel flow. The crossover operation is implemented based on cut and splice operations on the graph paths, whereas the mutation operation is just like other GAs. Very closed to this approach, the authors in [11] introduced a VLC-GA for a system that solves planning problems, i.e., finding the best sequence of actions to achieve given goals. Also, in this graph-based paradigm, Cruz-Piris et al. [12] developed a VLC-GA to solve a road traffic coordination multi-path problem, in which the chromosome is broken down into cells that encode path branches. In terms of complex chromosome, genetic programming (GP) [13] is an analogous evolutionary mechanism to search for optimal computer programs, but is limited to problem that can be represented in a tree-based structure.

Regarding sensor networks, many studies have been using GAs in different ways for optimization problems [14], but only few research attempts adopted VLC-GAs. In [15], Deif and Gadallah proposed a VLC-GA to solve a wireless sensor network deployment problem whose objectives are maximizing coverage and minimizing the deployment cost. In their study, the chromosome is composed of a variable number of integers that correspond to labels of deployment tuples in a pre-calculated set, where a tuple consists of three parameters, including sensor type, deployment point and orientation angle. Another study is introduced in [16], which employed a VLC-GA in the path design of a mobile sink going through fixed nodes in a wireless sensor network to collect data. The design objective is to balance the global network energy consumption and hence maximizing the network lifetime. The chromosome in that study is a list of integers representing the node identifiers.

It can be observed that, most of the studies as well as others in the literature that adopt VLC-GA are graph based, where a chromosome represents a path in the graph, and genes are of integer type. Furthermore, in those studies, although the chromosome length is variable, but is limited to a specific range, hence the number of available combinations is also limited. In contrast, the VLC-GA proposed in this study makes use of real-typed genes, and the chromosome has no limit in length.

To apply metaheuristic algorithms, it is also necessary to have a simulator for sensor networks that is able to predict the energy characteristics of the nodes. There exists already a number of public sensor network simulation software such as NS (Network Simulator) [17], OMNET++ (Optical Micro-Networks Plus Plus) [18], OPNET (Optimized Network Engineering Tools) [19], JSIM (Java-based simulation [20], or technical software that supports sensor network simulation tools such as Matlab & Simulink. However, these simulators focus mainly on simulating the network structure, routing protocols and scenarios, but not caring about energy as well as time alive time of nodes in the network [21]. Thus, the problem of energy and energy status in the network node has not been well studied in simulation software. Therefore, in this study, a sensor network simulation platform is also developed and briefly introduced. The platform is capable of simulating network operations with scenarios for monitoring, managing, and coordinating energy in each node and the entire network to support in solving sensor network optimization problems.

The remainder of this paper is organized as follows: In Section II, the schedule optimization problem for sensor networks is introduced. After that, the proposed VLC-GA to solve the network schedule problem is formulated and explained in Section III, whereas the energy simulation platform for WSNs is briefly introduced in Section IV. In Section V, a case study with simulation results is given and discussed. Finally, concluding and perspective remarks are given in Section VI.

2. SENSOR NETWORK SCHEDULE OPTIMIZATION PROBLEM

Lowering power consumption is essential to ensure the longevity of sensor networks. This leads to many studies for optimization of the nodes' working schedule, besides efforts for seeking to use low-power electronic components. In this study, a general model of sensor nodes shown in Figure 1 is used. Regarding the energy, a node consists of three major components: a power unit, a storage unit, and consumers. The consumers include every element in the node that consumes energy, such as communication, processing and sensing elements. The power unit can be a harvesting module that helps the node to collect ambient energy, or to convert the energy from other external sources [22][23]. Finally, the battery unit is to store energy when not utilized directly. On the design basis, a consumer element may drain energy from the power unit or the battery, or both. It is also necessary to note that one of the power unit or the battery may be omitted, and when the power unit is present, the battery is usually rechargeable.

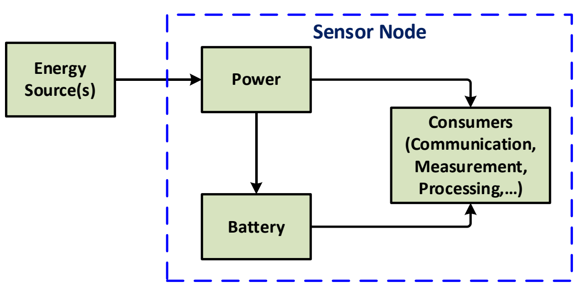

Fig. 1. Flow of energy in a typical sensor node.

Suppose the network to be scheduled is composed of n nodes indexed from 1 to n, which may be, but not necessarily identical. Each node, based on its own deployment conditions, may collect ambient energy at certain rate and time period. The goal is seeking to schedule each node to maximizing a specific goal of achievement provided by this network, while ensuring the energy-related constraints.

For a typical monitoring application, each sensor node in the network may function in active or sleep modes, each of them is characterized by a different average rate and pattern of power consumption. In active mode, the node can execute its programmed tasks such as sensing, measurement, data processing, data receival and transmission, etc., whereas in sleep mode, only a small low-power part of the node functions and carries out minimal activities to maintain the readiness for future activation. In more general cases, the nodes may have more than two modes. Denote the set of possible modes for node i as M^i, then a schedule S^i for node i is specified by a sequence of pairs:

$$S^i \triangleq \left\langle \left(m_j^i, t_j^i \right) \right\rangle \Big|_{j=1..s^i}, \tag{1}$$

where s^i is the length of this sequence, i.e., the number states for node i; $m_j^i \in M^i$ is the mode used in state j; and t_j^i is the starting moment of this state. In this study, for simplicity, only active-then-sleep sensors are considered, i.e., $M^i = \left\{ \bar{M}, \underline{M} \right\}$ for every node, where $\bar{M}$ is the active mode that corresponds to a higher power consumption level, whereas $\underline{M}$ is the sleep mode. Note that, even the nodes have the same modes, but they may have different power consumption characteristics. Also, the power consumption rates are not constant, but assumed to be predictable.

A schedule $\hat{S}$ of the whole network is just the combination of the schedules for every node, i.e.,

$$\hat{S} = \left\{ S^i \right\} \Big|_{i=1..n}. \tag{2}$$

Eventually, $t_1^i = 0$ and $t_{s^i}^i = T$ for every node, where T is the end time of the schedule. Theoretically, there is no limit for the number of states of one sensor node. This makes the schedule search space become very large. Therefore, the dependency of the objective function on the schedule is highly non-trivial and not suitable to be modelled or solved by traditional methods such as analytical ones. Preferable alternative methods are usually related to heuristics, such as machine-learning-based or evolutionary ones.

3. VARIABLE-LENGTH CHROMOSOME GENETIC ALGORITHM FOR NETWORK SCHEDULING PROBLEM

In this section, the solution of the network scheduling problem using a proposed VLC-GA is introduced with the theoretical formulation and implementation techniques of the algorithm. For the network scheduling problem, the chromosome C needs to encode the network schedule $\hat{S}$. In this study, it is defined as

$$\begin{aligned} C = \big[& s^1, m_1^1, t_1^1, m_2^1, t_2^1, \ldots, m_{s^1}^1, t_{s^1}^1, \\ & s^2, m_1^2, t_1^2, m_2^2, t_2^2, \ldots, m_{s^2}^2, t_{s^2}^2, \\ & \ldots, \\ & s^n, m_1^n, t_1^n, m_2^n, t_2^n, \ldots, m_{s^n}^n, t_{s^n}^n \big]. \end{aligned} \tag{3}$$

The number of genes for node i is $\left(2s^i + 1\right)$, and for the whole network is $\left(2\sum_{i=1}^{n} s^i + n\right)$. As s^i are variable, the length of C is also variable. It is important to note that in a VLC-GA, the fitness function is not a mathematical function in common sense, but rather a programmatical function, with takes the chromosome as input and returns the fitness value as output, after an execution process. In contrast to fixed-length chromosomes where the genes are located at fixed location, in a VLC, they are not. For this reason, s^i are included in C so that computer programs can interpret the genes correctly, otherwise it is not possible to know where the schedule information for each node starts and ends.

Basically, as C varies from individual to individual (remind that an individual corresponds to a network schedule configuration) and from generation to generation, then to express its dependency on the individual, denote ${}^{q}C(k)$ the chromosome of individual q in generation k. Similarly, all other individual-dependent variables also follow this notation convention as well, e.g., ${}^{q}m_j^i(k)$ is state j of node i in individual q in generation k.

Also, denote ${}^{q}C^i$ the segment in ${}^{q}C$ that correspond to the schedule of node i, that is,

$$ {}^{q}C^i = \left[{}^{q}m_1^i, {}^{q}t_1^i, {}^{q}m_2^i, {}^{q}t_2^i, ..., {}^{q}m_{{}^{q}s^i}^i, {}^{q}t_{{}^{q}s^i}^i \right]. \tag{4} $$

Note that the length of ${}^{q}C^i$ is $2\,{}^{q}s^i$, and ${}^{q}C$ can now be simplified as

$$ {}^{q}C = \left[{}^{q}s^1, {}^{q}C^1, {}^{q}s^2, {}^{q}C^2, ..., {}^{q}s^n, {}^{q}C^n \right]. \tag{5} $$

For visualization, ${}^{q}C^i$ can be illustrated graphically as shown in Figure 2. The mode switching events made by node i are presented along the time axis, where the higher-level segments correspond to the periods when the node is active, whereas the lower ones represent the periods when the node is in sleep mode.

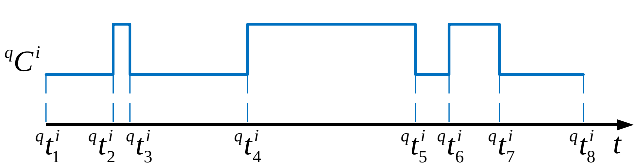

Fig. 2. Graphical Representation of a Node Schedule.

To implement a VLC-GA, it is necessary now to adapt the reproduction operations. The detailed implementation is explained in the remaining of this section. Obviously, the Selection operation does not need any modification for adaptation, only the two other operations do.

Regarding the Crossover operation, since in principle, each sensor node in a network may perform a different role, mating two different nodes has very little interest but would severely increase the algorithm complexity. Therefore, in this study, the mating is performed on the node basis. More clearly, the Crossover operation is applied to the chromosomes from the same node in the two individuals, but not others. In other applications, one can easily go further with cross-node mating as extension, if particularly interested. At this point, the Crossover operation on individuals is broken down to that on corresponding nodes.

The principle of the Crossover operation is illustrated by the example shown in Figure 3, where ${}^{q_1}C^i(k)$ and ${}^{q_2}C^i(k)$ are the parent chromosomes of a same node i but from two different individuals q_1 and q_2 in generation k, and ${}^{q_3}C^i(k+1)$ is the resulting child chromosome of the same node in the next generation $(k+1)$.

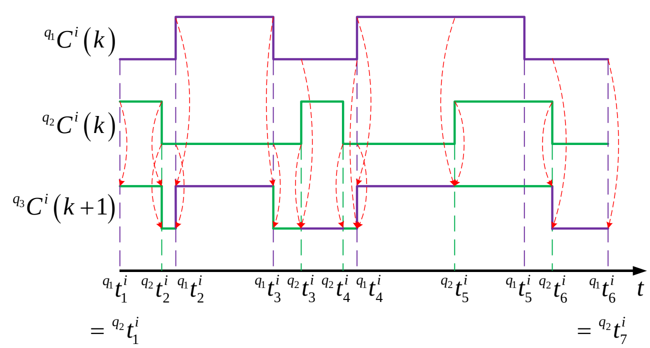

Fig. 3. Crossover Operation on Sensor Node Schedule.

To perform this operation, the timestamps are first sorted in increasing order with redundant ones excluded, so that the running time $[0,T]$ can be broken down into intervals. For the above example, the resulting time intervals are $\left[{}^{q_1}t_1^i, {}^{q_2}t_2^i\right]$, $\left[{}^{q_2}t_2^i, {}^{q_1}t_2^i\right]$, $\left[{}^{q_1}t_2^i, {}^{q_1}t_3^i\right]$, $\left[{}^{q_1}t_3^i, {}^{q_2}t_3^i\right]$, etc. After that, for each interval, the mode for the child node is determined by randomly picking a mode at the same interval from one of the two parents. In the figure, purple intervals in ${}^{q_3}C^i(k+1)$ represent the those inherited from ${}^{q_1}C^i(k)$, whereas green ones correspond to the those inherited from ${}^{q_2}C^i(k)$. To favor the genes from elite parents, higher probability can be associated to the modes from the parent with higher fitness value in the mode selection, and vice versa.

Finally, in the resulting chromosome, if two or more consecutive intervals have a same mode, they are combined for simplification. That is, $\left[{}^{q_1}t_3^i, {}^{q_2}t_3^i\right]$, $\left[{}^{q_2}t_3^i, {}^{q_2}t_4^i\right]$ and $\left[{}^{q_2}t_4^i, {}^{q_1}t_4^i\right]$ in the example are all in sleep mode and are simplified to $\left[{}^{q_1}t_3^i, {}^{q_1}t_4^i\right]$.

For Mutation operation, the principle is illustrated by the example shown in Figure 4. Starting from the parent chromosome $C^i(k)$, for each interval, one randomly chosen of the following four operations is carried out to produce the new one $C^i(k+1)$:

- *Copy*: The child interval is same as the parent interval, like the intervals $\left[t_1^i, t_2^i\right]$ and $\left[t_8^i, t_9^i\right]$ in the example.
- *Insertion*: A new interval with inverse mode is inserted in a manner that its start and end times are generated randomly but stay within the boundaries of the parent one, as in the case of the interval $\left[t_2^i, t_3^i\right]$ in the example.
- *Removal*: The parent interval is removed and not inherited by the child node, as in the case of the interval $\left[t_5^i, t_6^i\right]$ in the example.
- *Shift*: The boundaries of the child interval is made by moving those of the parent one backward or forward in the time axis, as in the case of the intervals $\left[t_3^i, t_4^i\right]$ and $\left[t_7^i, t_8^i\right]$ in the example.

These operations are performed at given rates ρ_M^C, ρ_M^I, ρ_M^R, ρ_M^S, corresponding to Copy, Insertion, Removal, Shift, respectively, where $\rho_M^C + \rho_M^I + \rho_M^R + \rho_M^S = 1$.

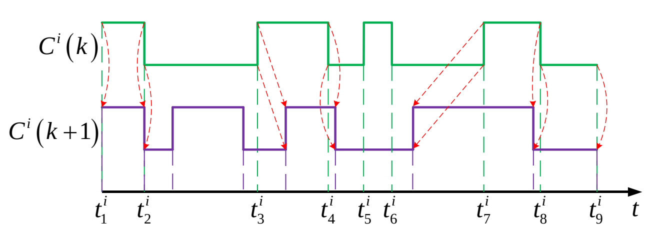

Fig. 4. Mutation operation on sensor node schedule.

For the example shown in the figure, the first interval $\left[t_1^i, t_2^i\right]$ is affected by a Copy operation, the second one $\left[t_2^i, t_3^i\right]$ is affected by an Insertion operation with a new subinterval randomly inserted in the middle, the third one $\left[t_3^i, t_4^i\right]$ is affected by a Shift operation with boundaries randomly moved rightward, the fifth one $\left[t_5^i, t_6^i\right]$ is affected by a Removal operation and disappears from the schedule.

The above mechanism of selection, crossover and mutation helps the algorithm to explore any possibility of network schedule by diversifying the generation of genes as the individuals evolve, and at the same time prioritize the good genes that make the corresponding individuals superior. Finally, it is necessary to mention that s^i elements in C are not subjected to the Crossover and Mutation operations, but are updated as a result of the application of these operations on corresponding C^i elements in C.

4. ENERGY-AWARE SIMULATION OF SENSOR NETWORKS

For any optimization method, one obligatory requirement is that, for any variable value $\mathbf{x} \in X$, it is possible to evaluate the objective function $f(\mathbf{x})$. Regarding our problem with sensor networks, in principle, to evaluate the fitness function, it is necessary to run the network in demanded conditions and measure the suitable parameters. However, heuristic methods like GAs usually need to evaluate the fitness function a huge number of times, hence the above fashion of fitness evaluation is just not realistic. To address this problem, a simulator is essential.

The workflow for schedule optimization is depicted in Figure 5. For given physical sensor nodes, their functioning mechanism as well as power-consumption characteristics are replicated by simulated nodes in the simulation platform. As stated earlier in this paper, the power consumption pattern in each mode of the sensor nodes is assumed to be predictable. Once the simulator is established, the proposed VLC-GA would be able to call it for any schedule and obtain back the required results for fitness evaluation. On termination of the genetic algorithm, the resulting optimal schedule is transferred and implemented into the physical nodes. The simulation platform is briefly introduced in remaining of this subsection.

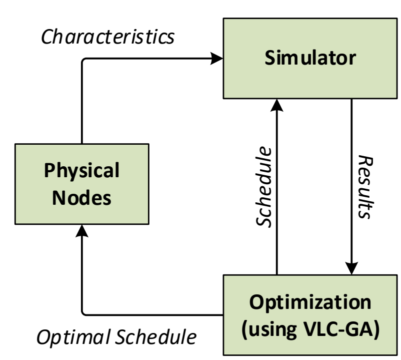

Fig. 5. Schedule optimization workflow.

The simulation platform is developed to allow to set up networks with high customization capabilities, set up the network's operating environment with environmental factors. Node is the basic element to construct the network structure. A typical sensor node consists of main components such as batteries, sensors, processing, energy harvesting and communication. From the above statement, the simulation platform developed in this study modeled the structure of the nodes with five main components as shown in Figure 6, including *Battery*, *Power*, *Sensor*, *Communication*, *Controller*.

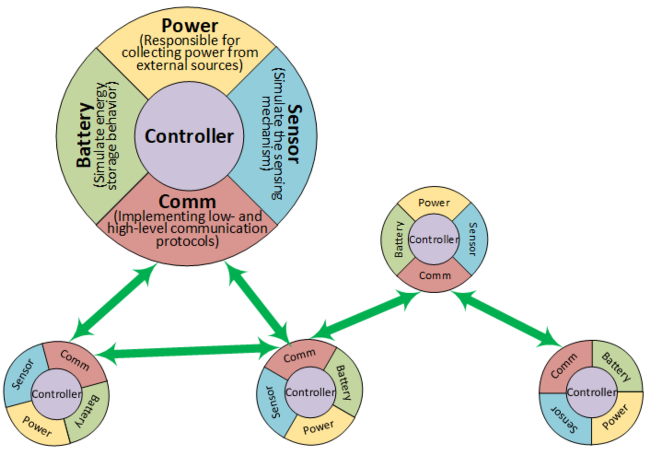

Fig. 6. Components of sensor nodes.

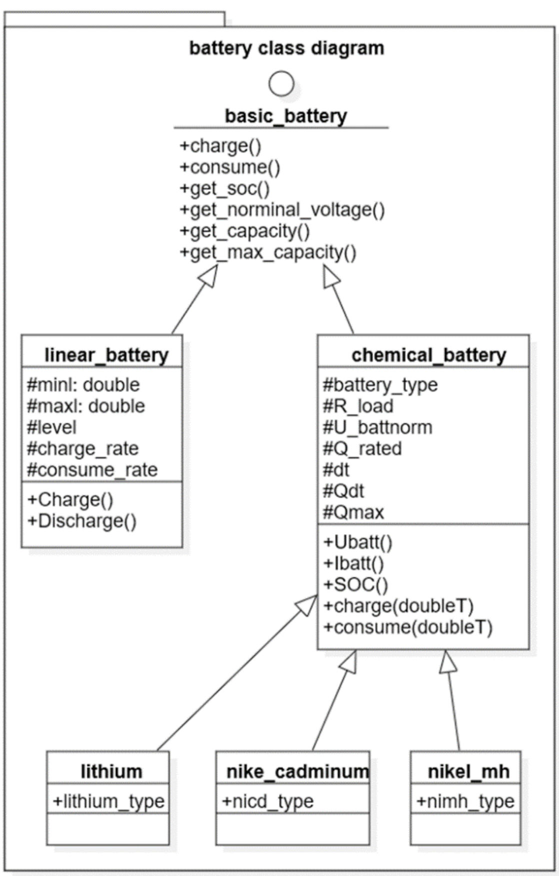

Fig. 7. Battery classes.

The network and sensor entities are designed with an event driven approach. Each entity can provide its own events, and allows other entities to register for notifications every time these events occur. An event that occurs at an entity is also propagated to the entities that own it, enabling events to be registered and captured by the same entity, but can also be registered at higher levels. To ensure the entities of the simulation system operate independently, each entity when activated will be allocated a separate thread to execute its operations without affecting others. Besides, each entity can set an arbitrary number of timers according to its requirements. Timers operate according to the local clock of the entity.

The *Control* block is responsible for processing and coordinating all the node operations according to the specific requirements of the scenario specified by the user. Building nodes in simulation software with 5 independent components will help the analysis and design of the toolkit become clearer and more flexible. Moreover, instead of users having to use nodes with a predefined structure, it is easy to set up nodes from these 5 independent components flexibly to have a node that matches its specific use purpose. This is also the difference and superiority compared to the current sensor network simulation tools. Besides that, a library of common battery types, including lead acid, Li-ion, Ni-Cd and Ni-MH, is implemented as part of this platform, as in the class diagram shown in Figure 7. More information about this simulation platform can be found in [24].

5. SIMULATION RESULTS

In this section, a scenario is designed to demonstrate the proposed VLC-GA. In this scenario, a network with three nodes labeled from 1 to 3 is considered to monitor certain environmental parameters within a period of 3 days with a daily schedule, i.e., the schedule for each day remains unchanged. Each of the nodes is designed with two modes, namely Active and Sleep, and consumes solar energy collected by a panel and is disposed with a battery to store redundant power. The solar radiation rate is variable of location on Earth and time in a manner that it is high in daytime and low in nighttime. It can be deduced based on the real solar position which is calculated with the algorithm provided in [25], where the assumed location on Earch is involved in the estimation of the solar radiation intensity by time in the day.

In Sleep mode, the nodes do not attempt to measure nor send any data, and only consume power at a minimal rate. On the other hand, when in Active mode, the nodes regularly measure the monitored environmental parameters, then sends the information to a station using SMS. All the measurement and transmission activities as well as the workload programmed in Active mode consume power. The principal parameters of the sensor node and the scenario are summarized in Table 1.

To make the nodes useful, we want to maximize the number of measurement values collected. However, as the battery can keep the nodes survive only for a certain time, they need to go in Sleep mode for specific periods. To keep the monitoring to be continuous enough, we also do not want that no measurements are performed for too long time. Finally, as the nodes need to work in days with similar conditions, its final battery level of each node in a day would not be lower than the initial one.

Table 1. Principal Simulation Parameters

Parameter	Value
Number of nodes (*n*)	3
Maximal battery capacity of node 1	3500mAh
Maximal battery capacity of node 2	5250mAh
Maximal battery capacity of node 3	7000mAh
Battery charging rate	0.8W
Background power consumption rate in Sleep mode	0.05W
Average power consumption rate in Active mode	0.17W
Power consumption per measurement	0.22Ws
Power consumption per SMS transmission	13.27Ws
Installation location (latitude, longitude)	21.004°, 105.846°
Measurement rate in Active mode	5 minutes
Simulation time (*T*)	3 days

With this design, the fitness function is defined with four components as follows:

$$\Phi = \Phi_1 + \Phi_2 + \Phi_3 + \Phi_4, \tag{6}$$

where Φ_1 is the term related to the number of measurements that we want to maximize, Φ_2 is used to penalize the long time periods when there no measurement is taken, Φ_3 is to penalize the schedule if the node is out of battery before the end of simulation, and Φ_4 is to penalize the schedule if the final battery level is lower than the initial one. More specifically,

$$\Phi_1 = -k_\eta \eta, \tag{7}$$

$$\Phi_2 = \sum_{i=1}^{\eta} k_{\tau 1} \Delta\tau_i + k_{\tau 2} \Delta\tau_i^2, \tag{8}$$

$$\Phi_3 = k_{T1}\left(T - \tilde{T}\right) + k_{T2}\left(T - \tilde{T}\right)^2, \tag{9}$$

$$\Phi_4 = \begin{cases} k_{L1}\left(L_e - L_s\right) + k_{L2}\left(L_e - L_s\right)^2 \text{ if } L_e < L_s, \\ 0 \text{ if } L_e \geq L_s, \end{cases} \tag{10}$$

where η is the total number of performed measurements; $\Delta\tau_i$ is the time difference between two consecutive measurements; $\tilde{T}$ is the time moment when the battery runs out, or is equal to T if this does not happen; L_s and L_e are initial and final battery levels, respectively; and k_η, $k_{\tau 1}$, $k_{\tau 2}$, k_{T1}, k_{T2}, k_{L1}, k_{L2} are constant weights. Note that quadratic functions are used in Φ_{2-4} to help the algorithm converges more quickly. The values of principal parameters used in the VLC-GA are given in Table 2.

Table 2. Principal Parameters of VLC-GA

Parameter	Value
Population size	100
Selection rate	20%
Crossover rate	50%
Rates of mutation operations: Copy, Insertion, Removal, Shift	90%, 3.33%, 3.33%, 3.33%
k_η	1
$k_{\tau 1}$, $k_{\tau 2}$	1, 10
k_{T1}, k_{T2}	1×10^8, 1×10^8
k_{L1}, k_{L2}	1×10^6, 1×10^6

Figure 8 shows the best fitness value, which decreases in an exponential-like manner to 3.680×10^4, which corresponds to the network schedule shown in Figure 9. By the obtained optimal schedule, the active time coverage in a day for the three nodes individually are 41%, 46% and 43%, but their combination makes a coverage of 95% of time in a day. The battery capacity percentage of the nodes when simulated with this schedule is given in Figure 10, and the cumulated number of measurements is given in Figure 11, which is now smoother than that in the previous scenario. At the end of simulation, the nodes perform 315, 351 and 332 measurements individually, or 998 in total. The longest interval without measurement from any node is 0.74h, while for each node individually is 9.3h, 5.7h and 6.3h, respectively.

With the initial and collected solar energy, the nodes are not able to cover all time in the day and the schedule is distributed to maximize the number of measurements. From this figure, it can be observed that the network is able to perform the measurements smoothly, even each node individually is not able to achieve that. The combined measurements are distributed uniformly over time, with the number keep increasing gradually.

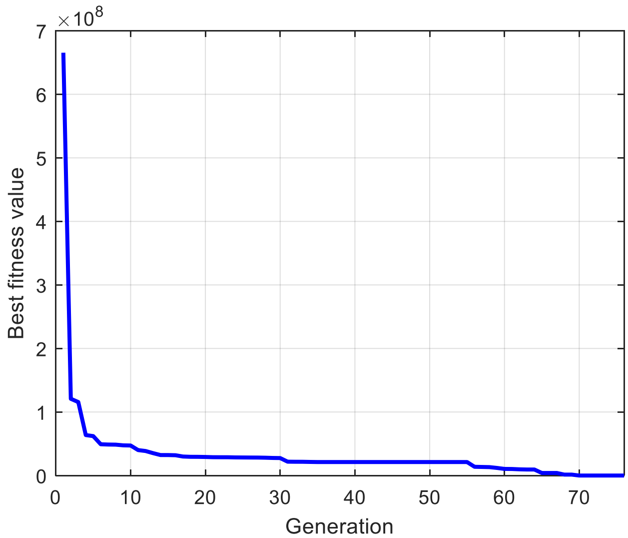

Fig. 8. Best fitness value.

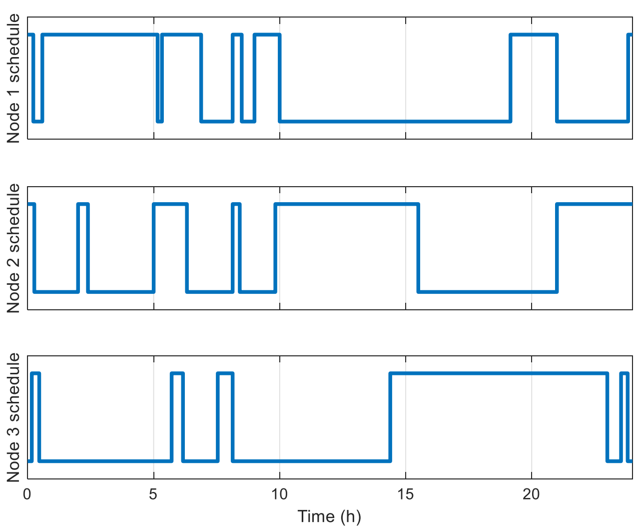

Fig. 9. Best network schedule.

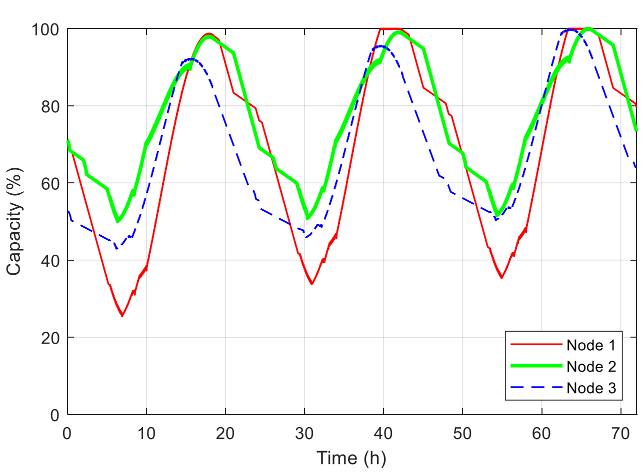

Fig. 10. Battery capacity of best sensor node.

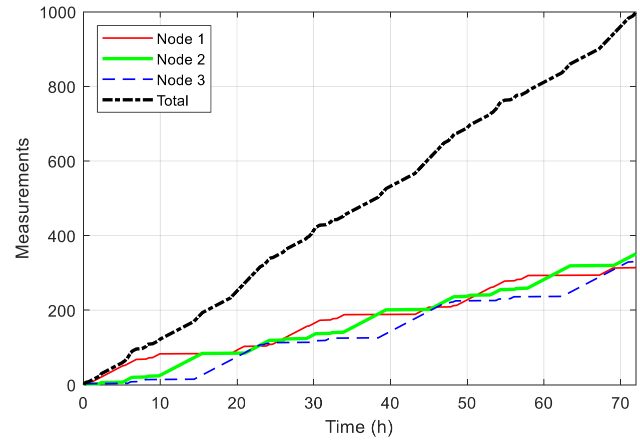

Fig. 11. Number of Measurements by Time.

6. CONCLUSION

For deployment of sensor networks, schedule optimization is critical and is a non-trivial problem. The test cases presented in this paper are rather simple, but they can clearly show that the proposed adaptation scheme for genetic algorithms with variable-length chromosome is an efficient and promising method to address this problem. It would also be possible to extend this work furthermore for application in sensor nodes with more than two states without difficulties, or even with dynamic schedules in which each node may change its state not only based on time, but also on other internal or external conditions.

While the proposed technique has been proved for its ability to extend the representation space of GAs, its performance may be problematic as the search space is too large. The complexity of the proposed algorithm would be critical when the number of sensor nodes is higher. To address this problem, performance-improving techniques should be considered in future work. Promising techniques include surrogate-assisted interactive GA proposed by Sun et al. [26], parameter adaptation and fitness-based neighborhood proposed by Gao et al. [27], distributed direction information-based mutation operators proposed by Peng et al. [28].

ACKNOWLEDGEMENT

This work was supported by the Vietnam Ministry of Education and Training under project grant number CT2020.02.BKA.03.

REFERENCES

[1] El Emary, I. M., & Ramakrishnan, S. (2013). Wireless Sensor Networks: From Theory to Applications. CRC press.

[2] Deng, J., Han, Y. S., Heinzelman, W. B., & Varshney, P. K. (2005). Scheduling sleeping nodes in high density cluster-based sensor networks. *Mobile Networks and Applications*, 10(6), 825-835.

[3] Wang, L., & Xiao, Y. (2006). A survey of energy-efficient scheduling mechanisms in sensor networks. Mobile Networks and Applications, 11(5), 723-740.
[4] Miller, M. J., & Vaidya, N. H. (2005). A MAC protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Transactions on mobile Computing*, 4(3), 228-242.
[5] Nawaz, F., Hassan, S. A., Hashmi, A. J., & Jung, H. (2019). A physical-layer scheduling approach in large-scale cooperative networks. *IEEE Access*, 7, 134338-134347.
[6] Nguyen, M. T., Nguyen, T. T., & Teague, K. A. (2020). An Energy-Efficient Combination of Sleeping Schedule and Cognitive Radio in Wireless Sensor Networks Utilizing Compressed Sensing. *In International Conference on Engineering Research and Applications* (pp. 154-160). Springer, Cham.
[7] Azimi, S. and Keyhan, K. (2016). The design a fuzzy system in order to schedule sleep and waking of sensors in wireless sensor networks. *International Journal of Engineering Research and Applications.* Vol. 6, no. 4, pp. 37-42, Apr.
[8] Daneshvaramoli, M., Kiarostami, M. S., Monfared, S. K., Karisani, H., Dehghannayeri, K., Rahmati, D., & Gorgin, S. (2020). Decentralized Communication-less Multi-Agent Task Assignment with Cooperative Monte-Carlo Tree Search. *In 2020 6th International Conference on Control, Automation and Robotics (ICCAR)* (pp. 612-616). IEEE.
[9] Holland, J. (1975). Adaptation in natural and artificial systems (University of Michigan press, Michigan).
[10] Jiang, H., Yang, X., Yin, K., Zhang, S., & Cristoforo, J. A. (2011). Multi-path QoS-aware web service composition using variable length chromosome genetic algorithm. *Information Technology Journal*, 10(1), 113-119.
[11] Brie, A. H., & Morignot, P. (2005). Genetic Planning Using Variable Length Chromosomes. *In ICAPS* (pp. 320-329).
[12] Cruz-Piris, L., Marsa-Maestre, I., & Lopez-Carmona, M. A. (2019). A variable-length chromosome genetic algorithm to solve a road traffic coordination multipath problem. *IEEE Access*, 7, 111968-111981.
[13] Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). Genetic programming: an introduction (Vol. 1). San Francisco: Morgan Kaufmann Publishers.
[14] Kaur, H., Tejpal, G., & Sharma, S. (2017). A review article on genetic algorithm in wireless sensor network. *In 2017 2nd International Conference on Communication and Electronics Systems (ICCES)* (pp. 246-250). IEEE.
[15] Deif, D. S., & Gadallah, Y. (2014). Wireless sensor network deployment using a variable-length genetic algorithm. *In 2014 IEEE wireless communications and networking conference (WCNC)* (pp. 2450-2455). IEEE.
[16] Anwit, R., & Jana, P. K. (2018). A variable length genetic algorithm approach to optimize data collection using mobile sink in wireless sensor networks. *In 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)* (pp. 73-77). IEEE.
[17] Issariyakul, T., & Hossain, E. (2009). Introduction to Network Simulator 2 (NS2). *In Introduction to Network Simulator NS2 (pp. 1-18). Springer, Boston, MA.*
[18] Xian, X., Shi, W., & Huang, H. (2008). Comparison of OMNET++ and other simulator for WSN simulation. *In 2008 3rd IEEE Conference on Industrial Electronics and Applications.* (pp. 1439-1443). IEEE.
[19] Chen, M., Miao, Y., & Humar, I. (2019). Introduction to OPNET Network Simulation. *In OPNET IoT Simulation (pp. 77-153). Springer, Singapore.*
[20] Neves, P. A., Veiga, I. D., & Rodrigues, J. J. (2008). G-JSIM—a GUI tool for Wireless Sensor Networks simulations under J-SIM. *In 2008 IEEE International Symposium on Consumer Electronics* (pp. 1-4). IEEE.
[21] Siraj, S., Gupta, A., & Badgujar, R. (2012). Network simulation tools survey. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(4), 199-206.
[22] Soyata, T., Copeland, L., & Heinzelman, W. (2016). RF energy harvesting for embedded systems: A survey of tradeoffs and methodology. *IEEE Circuits and Systems Magazine*, 16(1), 22-57.
[23] Shaikh, F. K., & Zeadally, S. (2016). Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 55, 1041-1054.
[24] Ha, V.P., Dao, T.K., Le, M.H., Nguyen, T.H., and Nguyen, V.T. (2020). Design and implementation of an energy simulation platform for wireless sensor networks. *2020 IEEE International Conference on Multimedia Analysis and Pattern Recognition (MAPR.* Hanoi, Vietnam, Oct.
[25] Michalsky, J. J. (1988). The astronomical almanac's algorithm for approximate solar position (1950–2050). *Solar Energy*, 40(3), 227-235.
[26] Sun, X., Gong, D., Jin, Y., & Chen, S. (2013). A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning. *IEEE Transactions on Cybernetics*, 43(2), 685-698.
[27] Gao, W., Chan, F. T., Huang, L., & Liu, S. (2015). Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood. *Information Sciences*, 316, 180-200.
[28] Peng, Z., Liao, J., & Cai, Y. (2015). Differential evolution with distributed direction information-based mutation operators: an optimization technique for big data. *Journal of Ambient Intelligence and Humanized Computing*, 6(4), 481-494.