Engineering Applications of Artificial Intelligence 65 (2017) 137–146

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Game theory based node scheduling as a distributed solution for coverage control in wireless sensor networks

Mahdi Movassagh, Hadi S. Aghdasi *

Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran

ARTICLE INFO

Keywords:
Area coverage problem, Game theory, Network lifetime prolonging, Distributed node scheduling methods, Wireless sensor networks

ABSTRACT

One of the important quality of services of Wireless Sensor Networks is the coverage, which focuses on providing near optimum coverage rate without decreasing network lifetime. Many distributed solutions have been proposed for this issue most of which try to select a minimum number of nodes as active while keeping others in sleep mode to preserve energy and extend network lifetime. However, these methods suffer from lacking a mathematical basis for their nodes selection approach. In this paper, we propose a distributed method for tackling this challenge by exploiting Game Theory as the mathematical basis for selecting active nodes named Game Theory based node Scheduling for Coverage control (GTSC). In GTSC, nodes compete each other to become active through exploiting their coverage redundancy, activation cost, the number of active neighbors, and uncovered region. The comparison of simulation results with the results of a well-known method and a state-of-the-art one shows that the proposed method outperforms both of them in terms of prolonging coverage, network lifetime, and energy efficiency, besides the redundancy rate reduction.

1. Introduction

For the last two decades, low cost and small size sensor nodes have attracted a lot of market and researchers' attention (Min et al., 2015; Wang et al., 2006). These devices usually are deployed in large numbers to form a network of cooperative nodes known as Wireless Sensor Networks (WSNs) (Assad et al., 2016). WSNs are used for wide range of applications, such as in agriculture, health care, industry, battlefield, and disaster management (Min et al., 2015; Wang et al., 2006; Wu and Cardei, 2016; Kisseleff et al., 2016; AlSkaif et al., 2015; Latif et al., 2016; Guo et al., 2015; Chen et al., 2016; Alkhatib, 2016; Li and Wu, 2016). In general, the basic idea is that each node collects regional information and sends them hop-by-hop to the sink. In most of the WSNs applications, it is infeasible to replenish the nodes' batteries. Thus, among WSNs constraints in memory, processing, and etc. energy constraint is the most important one. This makes new challenges for researchers to make customized or even new algorithms and methods for different tasks of WSNs like communication, processing, deployment, and so on (Hao et al., 2016).

In most of the mentioned applications, expected Quality of Services (QoSs) like maintaining coverage and connectivity besides the constrained battery capacity of nodes create challenges for researchers. One of the important challenges is the coverage and maintaining it at an acceptable rate while extending the network lifetime (Zhang and Hou, 2005). Knowing that usually nodes are deployed randomly and it is infeasible to replenishing their batteries, it is more acceptable to deploy nodes densely for satisfying mentioned applications (Shih et al., 2001). Thus, the challenge is to have a set of nodes be activated in each round — known as cover set — while keeping redundant nodes in sleep mode. Different solutions for this challenge can be put into two main categories: Centralized and Distributed (Zhu et al., 2012). In centralized solutions (Jameii et al., 2016, 2015) the general state of all nodes are gathered in the central station (sink) so that it selects cover set. In contrary, in the distributed methods, nodes decide their mode (active or sleep) according to their local information. Although centralized methods give the optimum result, they need to gather information from all the nodes in each round and send back the results to them, which are time and energy consuming processes. Thus, we have chosen the latter category in the rest of the paper to tackle the coverage problem.

Many distributed solutions have been proposed for solving the coverage problem in WSNs. Zhang and Hou have proposed the well-known optimal geographical density control (OGDC) method (Zhang and Hou, 2005) in which nodes autonomously schedule themselves based on local information but it suffers from a high degree of randomness in the method. Le et al. proposed CESS (Le and Jang, 2015)

* Correspondence to: Computer Engineering Department, Faculty of Electrical and Computer Engineering, University of Tabriz, (51666-15813), Tabriz, Iran.
E-mail addresses: m.movassagh93@ms.tabrizu.ac.ir (M. Movassagh), aghdasi@tabrizu.ac.ir (H.S. Aghdasi).

http://dx.doi.org/10.1016/j.engappai.2017.07.020
Received 17 February 2017; Received in revised form 26 June 2017; Accepted 24 July 2017
0952-1976/

in which nodes schedule themselves cooperatively and based on the coverage contribution of their neighbors but besides problems with nodes' decision making it also uses poor coverage contribution detection mechanism. Ai et al. proposed a game theory based distributed method (Ai et al., 2007) to solve the problem by grouping nodes in cover sets which maximizes network lifetime while guaranteeing the expected coverage of the area. However, this method does not consider important parameters' calculation as a part of the method. Wang et al. proposed LDCC (Wang et al., 2009) in which nodes decide to go to off or on mode autonomously based on the hop counts to the Base Station (BS). While guaranteeing connectivity, this method can end in reduced coverage rate and increased number of blind points in the condition of low or medium density network or uncontrolled redundancy rate. Hefeeda and Ahmadi proposed PCP (Hefeeda and Ahmadi, 2007) which activates sensors to form virtual triangular lattices but cannot guarantee full coverage as a matter of random deployment.

In this paper, we aim to tackle coverage problem with a distributed method named Game Theory based node Scheduling for Coverage control (GTSC). In GTSC, nodes schedule themselves autonomously based on exchanging low-level local information of their neighbors with very low message passing cost. We expect that with low message passing we get better results comparing to state-of-the-art (CESS) method and well-known (OGDC) one due to using Game Theory as the mathematical logic of selecting initial nodes instead of random selection. In GTSC, nodes compete each other to become active through exploiting their coverage redundancy, activation cost, the number of active neighbors, and uncovered region. The simulation results show that the proposed method improves OGDC and CESS, up to 1.64 times in terms of network lifetime, 1.02 to 2.35 times in terms of coverage lifetime, 16% to 31% in terms of the number of active nodes, and 13% to 26% in terms of redundancy rate.

The rest of the paper is organized as follows. In Section 2 we give a short review of previous works for solving coverage problem in WSNs. Preliminaries and problem formulation are presented in Section 3. Section 4 is allocated for an explanation of proposed method. In Section 5, the simulation setup and results are presented. Finally, Section 6 concludes the paper.

2. Related works

Many distributed methods have been proposed in the field of WSNs coverage. Danratchadakorn and Pornavalai (2015) have developed CMSS in which the area is partitioned into cell grids and the main effort is to activate only one node in each cell. Each node exchanges information with its neighbors to establish neighbors' table, calculates cell-values for them and defines waiting times to decide its mode as sleep or active. Yang et al. (2015) have proposed a novel method for area coverage in which instead of a binary coverage (disk model) method, probabilistic coverage is used. Adulyasas et al. have used hexagonal shapes for investigating an optimum number of nodes to guarantee full connected coverage which ends in O-Sym method which optimizes the overlapping coverage of nodes to get efficient coverage (Adulyasas et al., 2015). Mamun et al. have proposed a method (Mamun, 2014) for node selection in which each node decides its mode based on the number of neighbors, residual energy, shared covering the area with neighbors, and selection repetition number. Byun et al. proposed a smart method (Byun and Yu, 2014) based on cellular automata using nodes' local interactions with its neighbors, called environmental state signaling, to help nodes decide their mode autonomously as active or inactive. Lu et al. after proving that optimum node scheduling is an NP-Hard problem (Lu et al., 2015), and giving a polynomial time constant factor approximation, have proposed MLCS scheme which uses Coverage and Data Collection Tree (CDCT) to schedule nodes in order to guarantee both coverage and connectivity.

A novel approach for distributed decision-making problems like in WSN problems is Game Theoretical approach. Game theory platform enables WSN researchers to look at nodes as real autonomous entities that compete (and may cooperate) for network resources (i.e. energy and/or bandwidth) (AlSkaif et al., 2015). For example, Karimi et al. have proposed a game theoretical approach (Karimi et al., 2014) for solving clustering problem. There are few methods using game theory for solving coverage problem (AlSkaif et al., 2015). Here we review some of them. Ai et al. have proposed a game theoretical approach called Distributed, Robust and Asynchronous Coverage (DRACo) method (Ai et al., 2007) in which each node decides to be in one of k cover sets based on its coverage metrics. They later extended their work (Ai et al., 2008). The main problem with Ai et al.'s two models (Ai et al., 2007, 2008) lies in the process of deciding the number of cover sets or k which is not a part of the proposed method while it has a deep effect on the results. He et al. proposed a game theory based method (He and Gui, 2009) that uses (Ai et al., 2008) to find maximum coverage set and then uses Minimum Layer Overlapping Subfields (MLOF) as nodes' utility function. The problem with this work has been considered its lack of scalability, which means the number of iterations for MLOF is not sub-linear to the number of sensor nodes.

Zhang et al. first finds and proves mathematically (Zhang and Hou, 2005) the optimal pattern of node deployment to guarantee both coverage and connectivity under the condition that communication range should be at least twice the size of sensing range. Then, they propose the well-known OGDC method in which each node schedules itself based on the local information and as close as possible to the optimum pattern. The advantage of OGDC is that it has the least possible message passing overhead and uses very simple calculations. However, the main problem with OGDC is the high degree of randomness. This is due to the fact that the initial node declaring itself as the active node triggers selection of other nodes. Thus, the optimality of whole the method depends on selecting an initial node(s) while it is selected randomly. This work is very close to our proposed method in principles but our proposed mathematical logic for node selection procedure outperforms OGDC.

A novel method called Coverage and Energy Strategy for WSN (CESS) (Le and Jang, 2015) is also studied to compare its efficiency with our proposed method. In this work Le et al. after setting up an energy consumption model for nodes, introduce a node scheduling method for both coverage and connectivity. In the initial round, each node finds its neighbors by exchanging coordinates. Then it decides to go to sleep mode if it is fully covered according to a special coverage contribution mathematical model. Then it sends sleep mode declaration message to neighbors. When it decides to go to sleep mode, it makes each neighbor rechecks its coverage contribution. In next rounds, each previously active node decides its mode according to its energy level: (1) it remains active if its energy level is normal, (2) but if its energy level is low (below threshold), it sends HELP message and waits for volunteering neighbors. If it is fully covered by its volunteering neighbors, it sends back a list of to be activated nodes from the list of volunteered neighbors. (3) If its energy level is critical, it sends DUTY message, which turns neighbor nodes' mode to initial one. The first problem with this method is with its model for calculating coverage contribution. The mathematical model used for this purpose does not give the rate of coverage and just shows if the node is redundant or not and even for this goal it is an approximate model and does not give an exact answer (Tezcan and Wang, 2007). Besides, the main problem with this method is that it gives the duty of selecting active nodes to the low energy nodes. While this procedure is another selection problem and has NP-complete difficulty, the authors do not suggest a method for solving this issue. We will show that this issue makes big problems for CESS.

3. Preliminaries and problem formulation

3.1. Parameter description

Here we discuss three parameters used in our formulas and their theoretical and practical features.

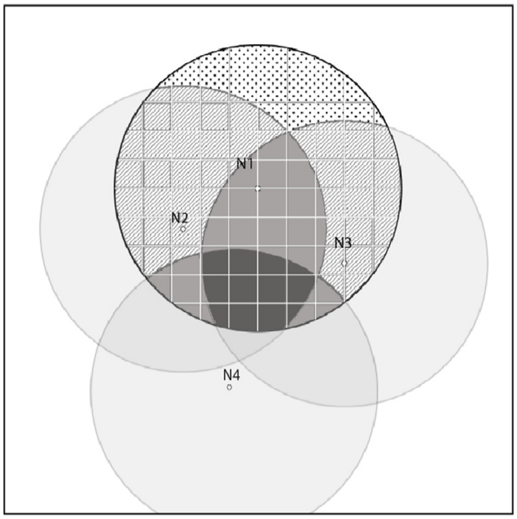

Fig. 1. Uncovered and redundant sub-regions.

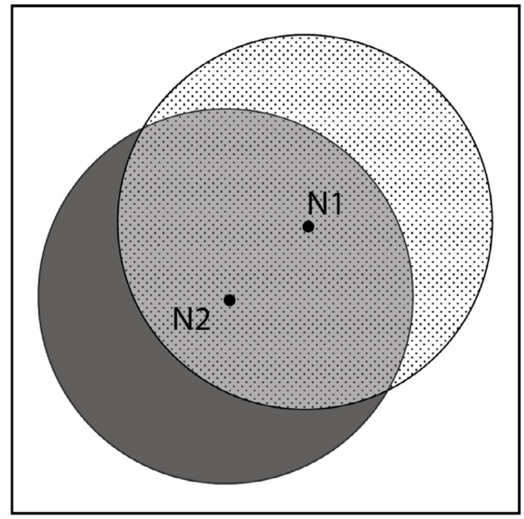

Fig. 2. Uncovered and redundant sub-regions.

3.1.1. Activation Cost (AC)

As sensing functions are energy-consuming tasks for nodes, following Declare as an Active node (DA) strategy should cost them to an extent. We call this amount of cost as Activation Cost (AC).

3.1.2. Uncovered Region (UR)

The main functionality and goal of WSN are to sense and monitor the Region Of Interest (ROI). Thus, having an uncovered sub-region in the sensing range of an inactive node by its neighbors can be considered as a cost for that node's payoff. The basic and general way of calculating this parameter for each node is to divide the ROI of the node into sub-regions and count the number of sub-regions, which are not in sensing range of the neighbor nodes and then normalize it by dividing to the number of all sub-regions. In Fig. 1, dotted sub-regions are uncovered sub-regions for N_1.

3.1.3. Redundant coverage

If a sub-region is monitored by more than one sensor node, redundant data will be generated and energy consumption, for both of the monitoring and data handling (communication, aggregation, etc.), will be increased. Thus, we consider this redundancy as the cost of DA strategy selection. The calculation of this parameter is done as the UR parameter except that unlike UR this parameter does not have upper bound. This feature can be seen in Fig. 1. While dashed sub-regions are 1-covered, Gray sub-regions are covered by more than one node and the color saturation shows the degree of redundancy for the sub-region. Gray-colored sub-regions are counted more than once in counting redundant sub-regions and this shows that no pre-defined upper bound can be considered for this parameter. In order to normalize this parameter to a number between 0 and 1 by investigating the maximum possible redundant sub-region:

$$RC_i = \frac{\left(\sum_{j=1}^{N_i} RCS_{i,j}\right)}{MRCS_N}. \tag{1}$$

In Eq. (1) RC_i is the RC parameter of sensor node i, and is a number between 0 and 1. $RCS_{i,j}$ is the number of sub-regions in common between nodes i and j. $MRCSN$ is the maximum redundant coverage sub-regions' count possible for a node in a network of N nodes.

3.2. Coverage game definition

We formulate the coverage problem as a non-cooperative game in which nodes are the players and compete in each round to be active or inactive with the aim of covering the whole area with a minimum number of active nodes. This game can be formulated as $CGT = \langle N, S, U \rangle$ in which CGT is the abbreviation for Coverage Game Theory, N is the set of players, S is the set of strategies, and U is the set of utility functions used to compute players' payoff.

The players set is the set of all nodes, which are distributed in the area uniformly. Each node at any given time can be in one of the two modes, Active or Sleep. So the set of strategies of each player are described as $S = \{DA, DI\}$ in which DA means "Declare as Active node" and DI means "Decide to be Inactive". As our aim is to cover the whole area with the minimum number of nodes and as minimum redundant coverage as possible. Therefore, the utility function can be described as below:

$$U_i = \begin{cases} 0 & \text{if } s_i \in DI \Lambda \nexists j \in N : s_j \in DA \\ v - CD_i & \text{if } s_i \in DI \Lambda \exists j \in N : s_j \in DA \\ v - CA_i & \text{if } s_i \in DA \end{cases} \tag{2}$$

$$CD_i = w_\alpha \times UR$$

$$CA_i = (w_\beta \times AC) + (w_\gamma \times RC).$$

In Eq. (2) s_i is the strategy of node i, v is a possible payoff, UR_i is the rate of sub-region of node i which is not covered by its neighbor nodes, RC_i is the sum of the rates of redundantly covered sub-regions of node i, and AC is the activation cost. If node i decides to go to sleep mode while it has no active neighbors, its ROI would be uncovered and thus its payoff will be zero. However, in case it has active neighbors that cover a sub-region of its ROI the payoff would be $v - CD_i$ in which v is the maximum payoff possible and CD_i (cost of inactive strategy) equals to the cost (UR). If node i decides to declare itself as active, its payoff would suffer the static cost of CA_i (cost of active strategy) as energy consumption and redundancy coverage in case there would be active neighbors.

The most desired situation is the second condition in Eq. (2) when CD_i (UR) is 0 and thus gets a possible payoff (v) and the worst case is the first one, which gets the payoff of 0. According to our considerations, the cost of activation while having redundancy from its neighbors is more than the cost of UR for an inactive node. In another word $CD_i < CA_i$.

For two nodes like Fig. 2, a table of payoffs based on their selected strategies can be built as Table 1.

Table 1
Payoffs of sensor nodes based on their following strategy.

Following strategy		*Payoffs*	
Player 1	*Player 2*	*Player 1*	*Player 2*
DI	*DI*	0	0
DI	*DA*	$v - CD_i$	$v - CA_i$
DA	*DI*	$v - CA_i$	$v - CD_i$
DA	*DA*	$v - CA_i$	$v - CA_i$

3.3. Nash equilibrium and activation probability

In a non-cooperative game, in order to find the probability of strategies' selection, the Nash equilibrium should be found. Table 1 shows that this game is a symmetrical game in which the payoff depends just on the players' strategies. In Table 1 $\langle DI, DI\rangle$ strategy is not Nash equilibrium because player prefers to play DA strategy to gain more payoff. $\langle DA, DA\rangle$ strategy is not Nash equilibrium also, because node prefers to play DI strategy to gain more payoff considering the fact that $CD_i < CA_i$. It can be concluded that either $\langle DI, DA\rangle$ or $\langle DA, DI\rangle$ is Nash equilibrium strategy. Also, the game's features suggest that there is no symmetrical Nash equilibrium.

As in this game, players compete with their direct neighbors, the above abstract table and strategy analysis can be generalized and extended to N players. Let $S = \{s1, s2, \ldots, sn\}$ denote the set of selected strategies by each node in an N node game. If all of them follow DI strategy, their payoff will be zero. If all of the players follow DA strategy, the payoff for each one will be $v - CA_i$. The third possible condition is that some of the players follow DA strategy with $v - CAi$ payoff while others follow DI strategy with $v - CDi$ payoff, which is the desired Nash equilibrium. As described before, CDi and CA_i depend on the situation of each node and its neighborhood and gives the dynamic needed to generate near-optimum results expected.

As the game is not symmetrical, nodes are allowed to play mixed strategies to have symmetrical Nash equilibrium (Osborne, 2004). This way we can find the probability of strategy selection by each node keeping in mind that we have just two strategies and the probability of DI strategy (q) is equal to $1 - p$, where p is the probability of DA strategy.

Theorem 1. *For the symmetrical coverage game, the probability of selecting DA strategy (p), by a node is equal to the symmetric mixed strategy Nash Equilibrium, which is given below:*

$$p = 1 - \left(\frac{AC + RC - UR}{v - UR}\right)^{\frac{1}{n-1}}. \tag{3}$$

Proof. First, we need to calculate the payoff for each strategy. As in DA strategy, node's payoff is independent of other nodes' strategies the expected payoff for DA strategy can be calculated as:

$$U_{DA} = V - w_\beta \times AC - w_\gamma \times RC. \tag{4}$$

But for DI strategy, we should consider neighbors' strategies. Thus, the payoff for this strategy can be calculated as:

$$\begin{aligned} U_{DI} &= P\{\text{none of neighbor nodes follow } DA\} \times 0 \\ &\quad + P\{\text{at least one neighbor node follows } DA\} \times \left(V - w_\alpha \times UR\right) \\ &= 0 + (1 - P\{\text{none of neighbor nodes follow } DA\}) \\ &\quad \times \left(V - w_\alpha \times UR\right) = \left(1 - q^{n-1}\right) \times \left(V - w_\alpha \times UR\right). \end{aligned} \tag{5}$$

In this context n is the number of nodes in the neighborhood (including the player node itself) which is a dynamic number (partial knowledge of node about its surrounding). Therefore, the payoff for DI strategy is:

$$U_{DI} = \left(1 - (1-p)^{n-1}\right) \times \left(V - w_\alpha \times UR\right). \tag{6}$$

Following the procedure described in Osborne's work (Osborne, 2004), as in equilibrium no one wants to change its strategy, to find the probability we put the two mentioned utilities equal. Now we can find the probability using equilibrium definition that no player wants to change its strategy:

$$\begin{aligned} &U_{DI} = U_{DA} \\ &\left(1 - (1-p)^{n-1}\right) \times \left(V - w_\alpha \times UR\right) = V - w_\beta \times AC - w_\gamma \times RC. \end{aligned} \tag{7}$$

Solving Eq. (7) will give the probability of following DA strategy as below:

$$p = 1 - \left(\frac{w_\beta \times AC + w_\gamma \times RC - w_\alpha \times UR}{v - w_\alpha \times UR}\right)^{\frac{1}{n-1}}. \tag{8}$$

Moreover, as the *AC*, *RC*, and *UR* are numbered in the range [0, 1) we can assume w_α, w_β, and w_γ simply as 1 which proves Theorem 1.

Theorem 2. *At least one node can get active in any kind of deployed networks.*

Proof. The range of the mentioned parameters suggest that *p* always lies between 0 and 1. Also as *n* increases, *p* decreases because the number of nodes that can cover the ROI increases. But the probability of at least one node's activation should not be zero:

$$\begin{aligned} p_1 &= P\{\text{at least one node gets active}\} \\ &= 1 - P\{\text{none of the nodes gets active}\} \\ &= 1 - q^n \\ &= 1 - (1-p)^n \end{aligned}$$

$$\overset{\text{Eq. (2)}}{\Rightarrow} p_1 = 1 - \left(\frac{AC + RC - UR}{v - UR}\right)^{\frac{n}{n-1}} \tag{9}$$

$$\lim_{n\to\infty} p = 0 \tag{10}$$

$$\lim_{n\to\infty} p_1 = 1 - \left(\frac{AC + RC - UR}{v - UR}\right). \tag{11}$$

From Eqs. (3) and (9) we can understand that if there is only one node, p and p_1 would be 1 and from Eqs. (10) and (11) we can understand that if *n* tends to infinity, p tends to 0 while p_1 tends to a number between 0 and 1 which depends on nodes deployment however it will not reach 0.

4. GTSC method

In this section, we describe the properties and procedures of Game Theoretical node Scheduling for Coverage control based on the logic discussed in Section 3. GTSC attempts to select nodes based on the coverage contribution that nodes can give in a competitive way. The whole procedure is done through completely local and without any extra information. In this way, the least possible message passing overhead will be imposed on the network while guaranteeing optimum coverage. First, we will give a short review of GTSC's assumptions and then discuss GTSC in details.

4.1. Overview and assumptions

GTSC needs basic information about the spatial relation between nodes. For this reason, we assume that nodes at least know their relative position to the area or their exact position. Considering the hardware available in the market and also the efficient localization techniques developed in recent years (Stanoev et al., 2016; Mittal et al., 2016; Lv et al., 2015; Jin et al., 2015) we believe this assumption is acceptable. However, as this method can work only using completely local information of its neighbors, the nodes can calculate and use the comparative and local position of its neighbors.

Each node assumed to have a communication range of at least twice the size of sensing range because with this assumption it is proved that full coverage guarantees full connectivity (Zhang and Hou, 2005).

Nodes are also assumed time-synchronized. This is important for nodes' active/sleep scheduling periods. This assumption is also considered to be feasible according to novel energy-efficient time synchronization techniques (Akhlaq and Sheltami, 2013; He et al., 2014).

GTSC in each round tries to select best possible nodes as active nodes according to the local information they get from their neighbors and then update their status. Each node can have four modes: *Volunteering*, *Listening*, *Active*, and *Sleep*. At the start of each round, all nodes start in *Volunteering* mode and calculate their probability to be the first node declaring itself as the active node. The nodes with higher energy are more probable to become starting node. Then all nodes receiving start node's message will go to *Competing* mode and calculate waiting time based on the relations discussed in the previous section, before declaring themselves as the active node. Each node in this mode updates its neighbor list and waiting time with each activation message they receive until its waiting time finishes or it is fully covered by its neighbors. At the end of GTSC operation, all nodes are in one of *Active* or *Sleep* modes.

4.2. GTSC details

The process of GTSC and its calculations for each node is based on its spatial relation with neighbors. However, in order for nodes to get aware of neighbors, there should be nodes that declare themselves as active without those spatial relations; we call these nodes as starting nodes. Having even one starting node on the condition of full connectivity is sufficient for GTSC to operate. Each node at the start of each round awakes in *Volunteering* mode. The transition between modes can be seen in Fig. 3.

Volunteering mode: In this mode, each node calculates a probability based on the equation below:

$$p_v = p_{rc} \times \left(1 - \frac{RE}{IE}\right). \tag{12}$$

Where *RE* is the residual energy of the node and *IE* is the initial energy of node. Selecting initial nodes using the probability in Eq. (12) guarantees that nodes with higher energy will be selected and thus the energy will decrease uniformly in whole the area. The p_{rc} in this equation is the randomness controller.

p_{rc} guarantees that among nodes with the same properties and status which and how many will be volunteered. If the node volunteers (based on the probability it calculates), it will wait for T_v, which is calculated as below:

$$T_v = T_d * \left(1 - \frac{RE}{IE} + \tau\right). \tag{13}$$

In Eq. (13) τ is used as a back-off timer to prevent simultaneous node activation and radio interference; Here, τ is uniformly distributed between 0 and 1. As it can be seen in Fig. 4, while waiting, the node listens to the channel for neighbors' activation message. If it receives a message before its timer ends, it will cancel its timer and go to *Competing* mode; but if its timer ends before any messages arrive, it will declare itself as an active node by sending activation message. Activation message simply contains just the node's coordinates.

Otherwise, if the node fails to volunteer, it will wait for T_{rv} (re-volunteering time) before re-calculating its probability; each time node repeats volunteering process, it doubles p_{rc} until it reaches 1. P_{rc} resets to an initial value (p_0) whenever node sleeps or activates. T_{rv} can be calculated as below:

$$T_{rv} = 2 * T_d * \left(1 - \frac{RE}{IE}\right). \tag{14}$$

Like the procedure mentioned above, if a node receives any message before its timer ends, it will change its mode to *Competing* mode. However, if its timer ends before receiving any messages it will redo the volunteering process with doubled p_{rc}.

Competing mode: In *Competing* mode, each time the node receives an activation message, adds the coordinate of the sender to its neighbor list and calculates a waiting time, T_c:

$$T_c = \left((1 - p) * AN_N + \tau\right) \times t_0. \tag{15}$$

In Eq. (15), *p* is the probability, which can be calculated using Eq. (8). The problem with Eq. (8) is that *n*, the number of neighbors, is a dynamic parameter and updates step by step. Therefore, there are two possible approaches for using this equation. The first approach is to use the latest information node has about its neighbors (the number of nodes that has sent activation message in current round). The second approach is to predefine the possible number of neighbors for each node in a deployment with a specific number of nodes. Here AN_N is a parameter that is defined as the average neighbor count possible for a node in an *N* node network. This parameter is obtained by simulation of random node deployment for 500 times. The same problem can be described for the redundancy rate normalization. In order to normalize redundancy rate, we introduce *MRCS*, *which* is the maximum redundancy possible for a deployment with a specific number of nodes. Like *AN*, *MRCS* is also pre-defined and is obtained through 500 times node deployment simulation for a different number of nodes.

Using *AN* beside the τ which is a random number of range 0 and 1 in Eq. (15), ensures that the waiting time of no two nodes in the neighborhood will be the same. t_0 is the average time for a message to be sent from sender to receiver.

During waiting time, if the node receives new activation message, it will update its neighbor list, reset its waiting timer and re-calculates the T_c. This procedure repeats until either the node is fully covered by its neighbors or its timer ends. In the first case node will go to sleep mode and in the latter case, node declares itself as an active node by sending the activation message.

5. Simulation results and analysis

In this section, first, we introduce the parameters tuning and assumptions done for simulation as simulation setup and then compare the simulation results of proposed method with related works.

5.1. Simulation setup

The simulations are done in an area of 50 × 50 m^2. Nodes are deployed randomly in the region. All nodes know their geographical or relative position of the area they have been deployed. Each node has a sensing range of 10 m and a connectivity range of 20 m and starts its work with the initial energy of 5 j. The nodes also have predefined parameters *AN* and *MRCS* prior to deployment which depend on the overall number of nodes deployed (*N*). The possible *AN* and *MRCS* values for an area of 50 × 50 m^2 can be obtained from Table 2.

For most of the parameters, we have used (Zhang and Hou, 2005). For timing, t_0, as described before, is the time needed for a message to be sent from sender to receiver and assumed to be 6.9 ms. T_d is about 1.5 times t_0 or simply 10 ms. In order to calculate *UR* and *RC*, each node divides its sensing region to grids of size 1×1 m^2 and checks the grids' center for coverage by its neighbors. Unlike OGDC in which each node requires the neighbors to fully cover all its inner grids to consider it as redundant, in GTSC each node requires a threshold of 90% of its inner grids to be covered by neighbors.

For energy consumption parameters, EC_t is described as the energy consumption rate for sending a message and is assumed to be 40 j/s. EC_i is the energy consumption rate in the idle state and is equal to 5 j/s. EC_s is the energy consumption rate for sensing and processing an event and is assumed to be 20 j/s. We assumed no energy consumptions for receiving messages and sleep mode of the nodes. Each active node in each round in addition to sensing 1000 events and sending it to the next hop relays 1000 messages generated by other nodes.

All the simulations are done in the Matlab 2016(a) and with the same general parameters in Table 3.

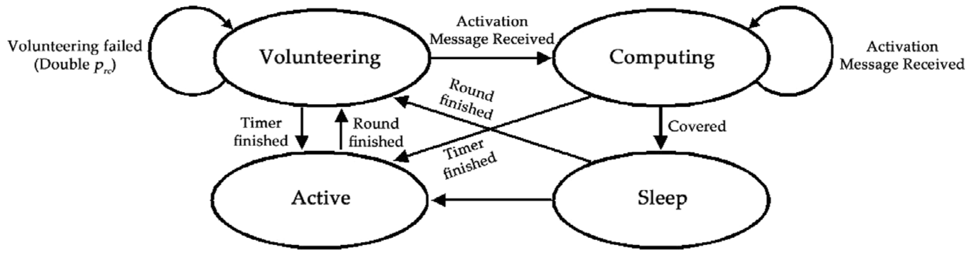

Fig. 3. The transition between different modes.

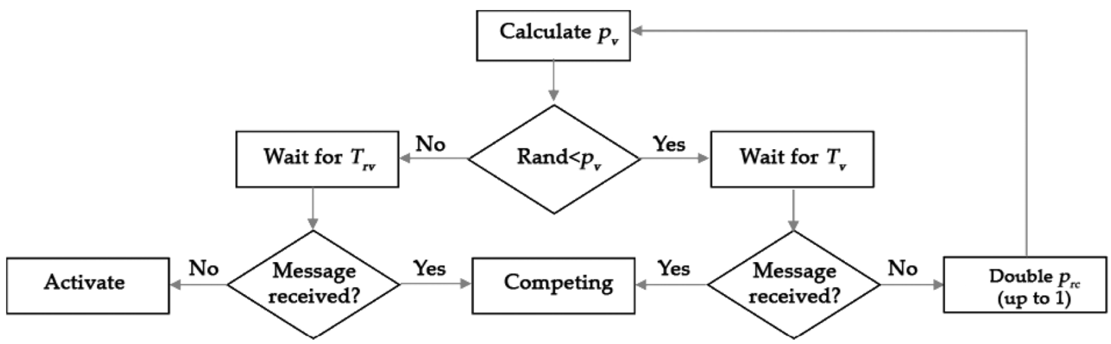

Fig. 4. Volunteering mode.

Table 2
Possible amount of AN and $MRCS$ for different numbers of "N" in 50 × 50 area.

N	*AN*	*MR*
20	7	258
30	10	380
40	13	494
50	17	612
60	20	725
70	24	843
80	27	953
90	31	1062
100	34	1174

Table 3
Parameter settings for simulation.

N	*AN*	*MR*
IE	Initial energy of node	5 j
p_0	Initial value of P_{rc}	K/N
K	The number of initial nodes desired	1
N	Number of deployed nodes	
t_0	Packet transmission time	6.9 ms
T_d	Time constant for volunteering	10 ms
TI	Time interval for each round of process	0.1 ms
Round time	–	1000 s
R_s	Sensing range of each node	10 m
R_c	Connectivity range of each node	20 m
EC_t	Energy Consumption rate for transmitting packets	40 j/s
EC_s	Energy Consumption rate for sensing events	20 /s

5.2. Simulation results

As mentioned before we will compare the simulation results of our proposed method with those of OGDC (Zhang and Hou, 2005) and CESS (Le and Jang, 2015) as famous and novel related works, respectively. To simulate these two methods, we have used their own parameters except for the energy parameters described here and compare them in the same simulation setup. We also investigate the behavior of proposed method when using a static number of average possible neighbor nodes as described in Section 4.2 (as a predefined parameter), compared to dynamic partial information of node about its neighbors. We have run the simulations with a different number of nodes deployed (20, 50, and 100 nodes) and repeated each simulation 20 times. As the number of rounds (Network lifetime) in each repeat is different, to get average results of all simulation repeats, we consider the maximum number of rounds achieved.

Coverage rate: The first important QoS compared among three methods is coverage rate. This QoS is expected to be high enough (almost 100%) as long as possible. Usually the coverage rate for coverage control methods start in a maximum of almost 100% but a method is considered more efficient if it can keep the coverage rate in high values for longer time. Thus, to compare this QoS we use the coverage lifetime. For example, the 90% coverage lifetime shows the duration of time (rounds) that the network provides at least 90% coverage rate by its active nodes. Fig. 5 show the achieved simulation results of the coverage rate.

As it can be seen in Fig. 5, GTSC can stay in high values of coverage rate for a longer time. The 90% coverage lifetime is improved up to 1.25 times for a different number of deployed nodes comparing to OGDC and for about 1.02–1.99 times compared to CESS in medium or highly dense deployments. For 80% coverage lifetime, GTSC outperforms OGDC by about 1.04–1.18 times and CESS by 1.10–2.35 times.

Coverage efficiency: Another factor for measuring the selection efficiency of proposed method is the coverage efficiency. Its idea is to consider the quality of node selection instead of considering the number of selected nodes. It shows the rate of coverage per maximum possible coverage which can be achieved by activating all the alive nodes. In other words, it shows how efficiently the active nodes are selected from the alive nodes. Simulation results in Fig. 6 show that proposed method can select optimum nodes efficiently longer than OGDC and CESS.

Active nodes: Another important factor is the number of active nodes, i.e. the minimum number of nodes guaranteeing full coverage, which should be activated to expand network lifetime.

Figs. 5 and 6 suggest that OGDC can reach the highest possible rate of coverage, however, Fig. 7 indicates that OGDC pays the cost of activating more nodes for this goal. The result is that both the network lifetime and α-Coverage lifetime is reduced.

CESS works optimally for as long as no node goes to Help mode in which it decides which volunteering neighbors should go to active mode.

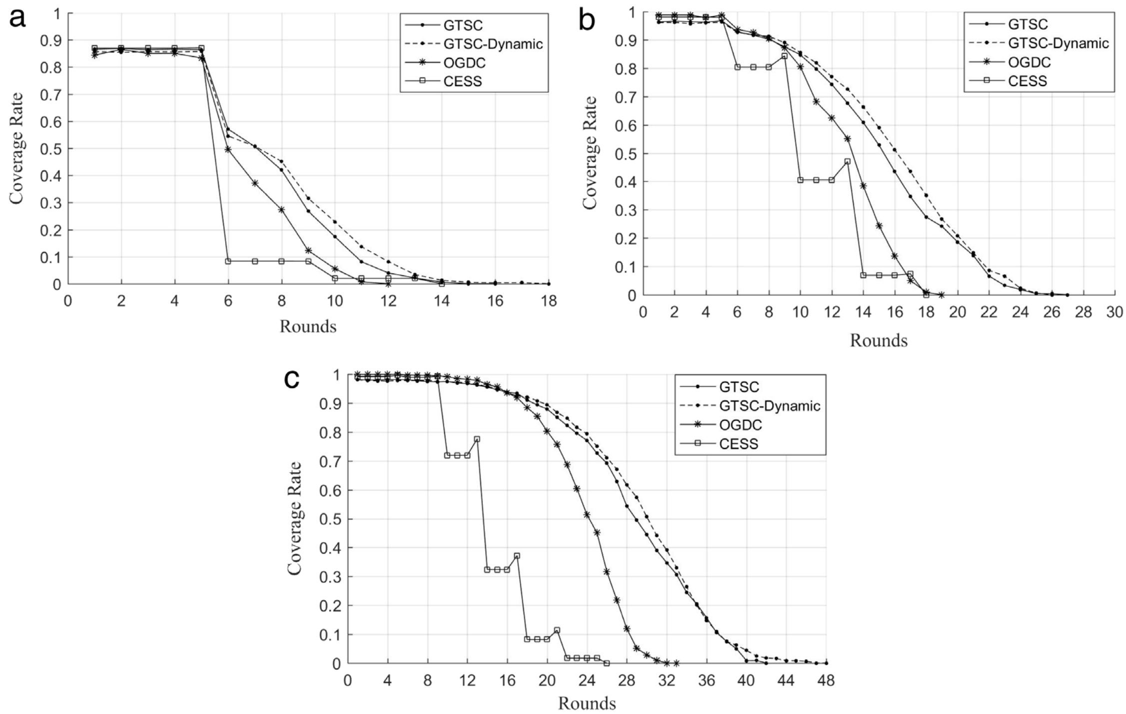

Fig. 5. Coverage rate, (a) 20 nodes, (b) 50 nodes, (c) 100 nodes.

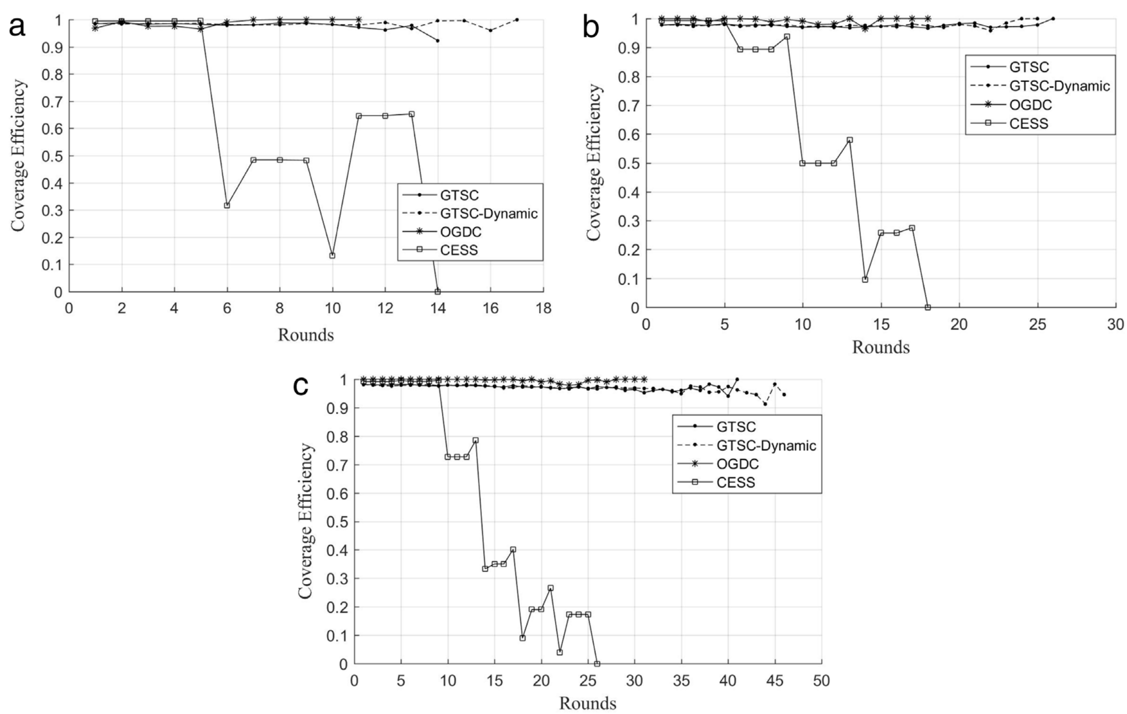

Fig. 6. Coverage efficiency, (a) 20 nodes, (b) 50 nodes, (c) 100 nodes.

As this decision-making is complex and time-consuming, these nodes use a simple method, which in return selects a non-optimum number of neighbors and thus, more nodes than needed become active.

All three methods cover the area with an average of 15 to 20 nodes (for a different number of deployed nodes) but in overall network lifetime, GTSC outperforms OGDC by 16% to 31% and CESS by 21% to 30%.

Redundancy rate: The next important factor for a coverage scheduling method is redundancy rate. The higher redundancy rate is, the higher redundant data will be generated and extra message passing overhead will be imposed to the network.

As it can be seen in Fig. 8, GTSC outperforms OGDC by about 13%–20% for different deployments during an 80% coverage lifetime. Also, it outperforms CESS by 24%–26% in the same coverage lifetime.

Total residual energy: The last measurement factor for the efficiency of the method is the overall residual energy of all the nodes deployed. This parameter can show how optimally and evenly the energy consumption rate is distributed in all nodes in different rounds.

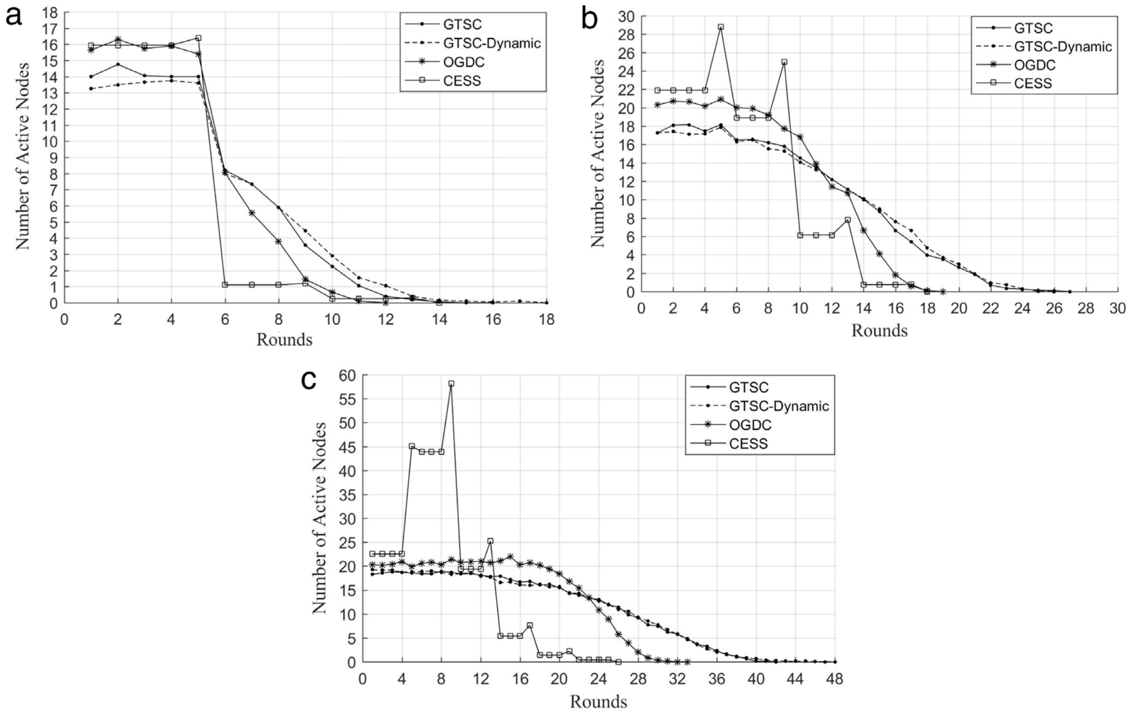

Fig. 7. Number of active nodes, (a) 20 nodes, (b) 50 nodes, (c) 100 nodes.

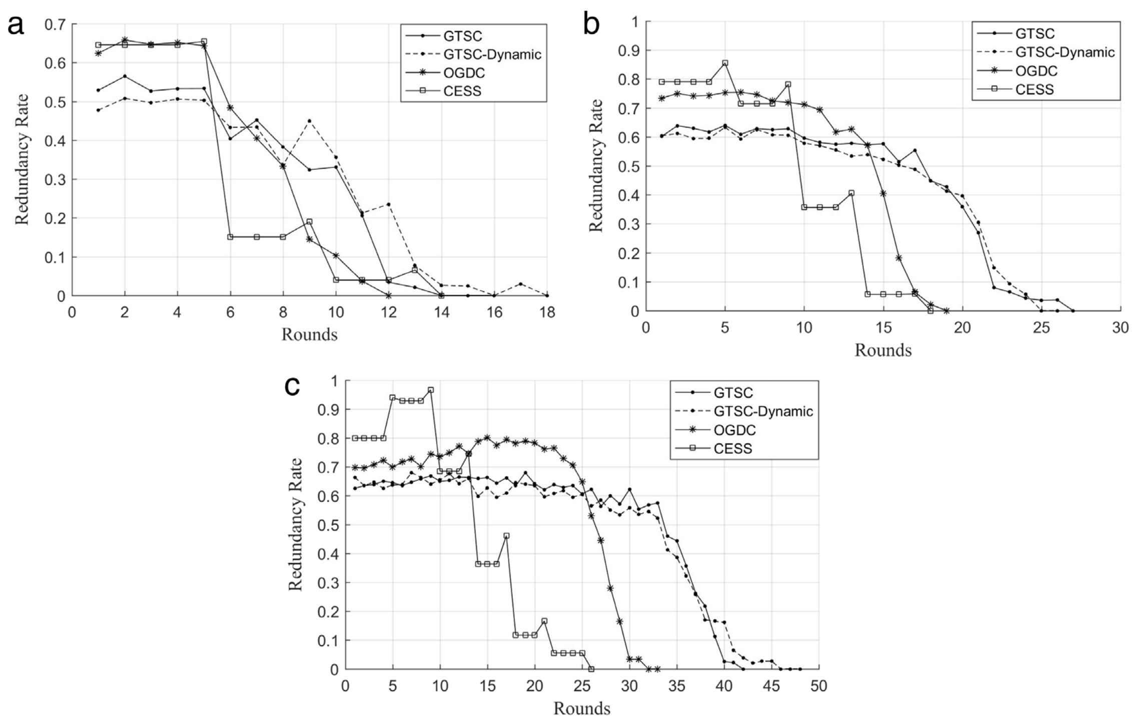

Fig. 8. Redundancy rate, (a) 20 nodes, (b) 50 nodes, (c) 100 nodes.

According to Fig. 9, the overall network lifetime is improved in the proposed method by 1.36–1.42 times compared to OGDC and 1–1.64 times comparing to CESS for a different number of nodes deployments.

As described before CESS works optimally until some nodes go to Help mode, and decide which volunteering neighbors should go to active mode. As this decision-making is complex and time-consuming, the basic selection method in use results in a non-optimum number of activated neighbors. With a significant increase in a number of active nodes, energy consumption rate increases and thus, nodes die earlier. Also, the residual energy does not reach 0 because no plan is described for nodes that their energy is less than a predefined threshold. For comparing purposes, we have considered this threshold the same as OGDC's one.

Taking Network lifetime, coverage efficiency, coverage rate, redundancy rate, and residual energy QoSs into consideration, it can be concluded that: (1) the proposed method is far better than state-of-the-art method CESS. (2) GTSC is almost as good as the near-optimum

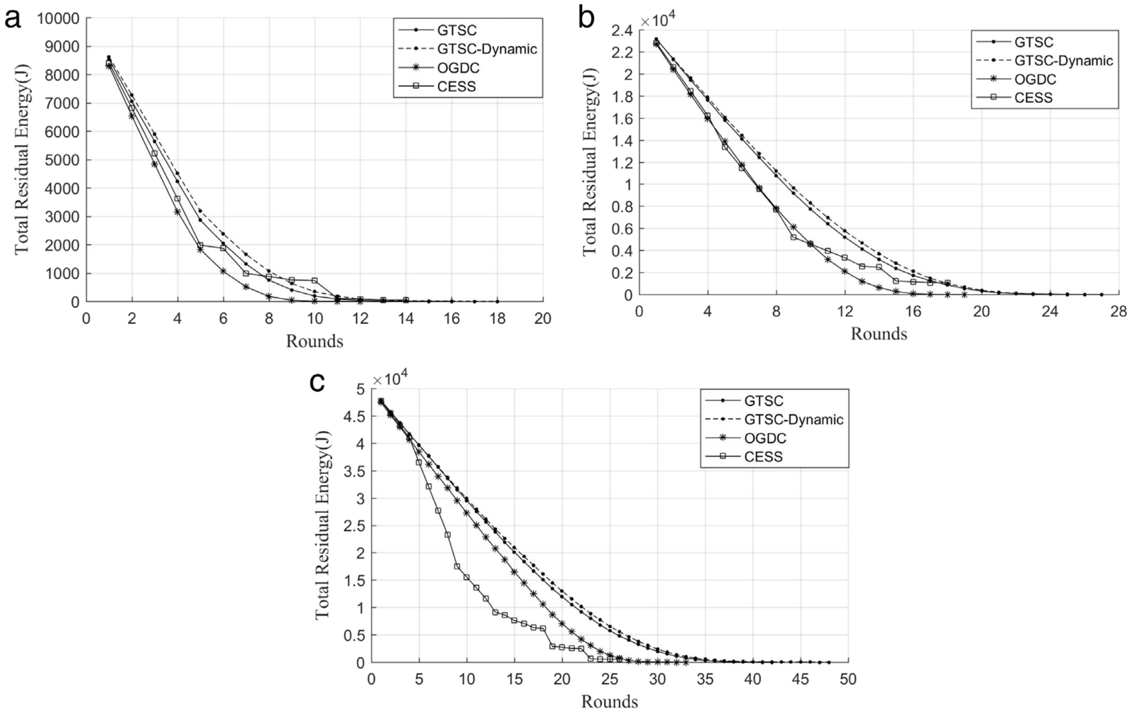

Fig. 9. Total residual energy, (a) 20 nodes, (b) 50 nodes, (c) 100 nodes.

OGDC and even better from Network lifetime point of view. (3) The different parameter tuning of GTSC for a number of neighbors has a very slight effect but from network lifetime point of view, static predefined maximum neighbor setting gives better results.

6. Conclusion

Wireless sensor networks have attracted computer science researchers because of the important role they can play in the industry, agriculture, security, healthcare, disaster management, and wildlife care applications. The sensor nodes can operate autonomously using their own hardware and software but their resources, especially their battery capacity, are limited. In this paper, we focused on the coverage QoS, which aims to prolong network lifetime while covering the whole area of interest. This can be achieved through selecting a minimum number of nodes to be activated in each round as cover set while preserving the network energy by keeping others in sleep mode.

After reviewing some of the related works proposed to tackle this problem, we concluded that most of them use greedy methods to select active nodes. Thus, we proposed GTSC. The main contributions of the proposed method are: (1) intrinsic features of Game Theory is used as the mathematical basis for selecting active nodes. (2) node scheduling is done completely distributed with local information. In this method, nodes are considered as the players of the cover game and compete to gain more profit of preserving energy and area coverage. The simulation results compared to well-known OGDC (Zhang and Hou, 2005) and state-of-the-art CESS (Le and Jang, 2015) proves the efficiency of the proposed method in terms of network lifetime, coverage, the number of activated nodes, and redundancy.

References

Adulyasas, A., Sun, Z., Wang, N., 2015. Connected coverage optimization for sensor scheduling in wireless sensor networks. IEEE Sens. J. 15, 3877–3892.

Ai, X., Srinivasan, V., Tham, C.-K., 2007. DRACo: Distributed, robust an asynchronous coverage in wireless sensor networks. In: 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks. SECON'07, IEEE, pp. 530–539.

Ai, X., Srinivasan, V., Tham, C.-K., 2008. Optimality and complexity of pure nash equilibria in the coverage game. IEEE J. Sel. Areas Commun. 26, 1170–1182.

Akhlaq, M., Sheltami, T.R., 2013. RTSP: An accurate and energy-efficient protocol for clock synchronization in WSNs. IEEE Trans. Instrum. Meas. 62, 578–589.

Alkhatib, A., 2016. Sub-network coverage method as an efficient method of wireless sensor networks for forest fire detection. In: Proceedings of the International Conference on Internet of Things and Cloud Computing. ACM, p. 65.

AlSkaif, T., Zapata, M.G., Bellalta, B., 2015. Game theory for energy efficiency in wireless sensor networks: Latest trends. J. Netw. Comput. Appl. 54, 33–61.

Assad, N., Elbhiri, B., Faqihi, M.A., Ouadou, M., Aboutajdine, D., 2016. On the deployment quality for multi-intrusion detection in wireless sensor networks. In: Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015. Springer, pp. 469–478.

Byun, H., Yu, J., 2014. Cellular-automaton-based node scheduling control for wireless sensor networks. IEEE Trans. Veh. Technol. 63, 3892–3899.

Chen, H., Li, X., Zhao, F., 2016. A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks. IEEE Sens. J. 16, 2763–2774.

Danratchadakorn, C., Pornavalai, C., 2015. Coverage maximization with sleep scheduling for wireless sensor networks. In: 2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, pp. 1–6.

Guo, W., Li, J., Chen, G., Niu, Y., Chen, C., 2015. A PSO-optimized real-time fault-tolerant task allocation algorithm in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 26, 3236–3249.

Hao, X., Liu, W., Yao, N., Geng, D., Li, X., 2016. Distributed topology construction algorithm to improve link quality and energy efficiency for wireless sensor networks. J. Netw. Comput. Appl..

He, J., Cheng, P., Shi, L., Chen, J., Sun, Y., 2014. Time synchronization in WSNs: a maximum-value-based consensus approach. IEEE Trans. Automat. Control 59, 660–675.

He, X., Gui, X., 2009. The localized area coverage algorithm based on game-theory for WSN. J. Netw. 4, 1001–1008.

Hefeeda, M., Ahmadi, H., 2007. A probabilistic coverage protocol for wireless sensor networks. In: 2007 IEEE International Conference on Network Protocols. IEEE, pp. 41–50.

Jameii, S.M., Faez, K., Dehghan, M., 2015. Multiobjective optimization for topology and coverage control in wireless sensor networks. Int. J. Distrib. Sens. Netw. 2015.

Jameii, S.M., Faez, K., Dehghan, M., 2016. AMOF: adaptive multi-objective optimization framework for coverage and topology control in heterogeneous wireless sensor networks. Telecommun. Syst. 61, 515–530.

Jin, R., Che, Z., Xu, H., Wang, Z., Wang, L., 2015. An RSSI-based localization algorithm for outliers suppression in wireless sensor networks. Wirel. Netw. 21, 2561–2569.

Karimi, N.B., Razavi, S.N., Aghdasi, H.S., 2014. Distributed clustering in wireless sensor networks unsing a game theoretical approach. Int. J. Tech. Phys. Probl. Eng. 6, 1–8.

Kisseleff, S., Chen, X., Akyildiz, I.F., Gerstacker, W.H., 2016. Efficient charging of access limited wireless underground sensor networks. IEEE Trans. Commun. 64, 2130–2142.

Latif, K., Javaid, N., Ahmad, A., Khan, Z.A., Alrajeh, N., Khan, M.I., 2016. On energy hole and coverage hole avoidance in underwater wireless sensor networks. IEEE Sens. J. 16, 4431–4442.

Le, N.-T., Jang, Y.M., 2015. Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks. Int. J. Distrib. Sens. Netw. 2015, 10.

Li, W., Wu, Y., 2016. Tree-based coverage hole detection and healing method in wireless sensor networks. Comput. Netw. 103, 33–43.

Lu, Z., Li, W.W., Pan, M., 2015. Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks. IEEE Trans. Veh. Technol. 64, 714–727.

Lv, X., Mourad-Chehade, F., Snoussi, H., 2015. Decentralized localization using radio-fingerprints and accelerometer in WSNs. IEEE Trans. Aerosp. Electron. Syst. 51, 242–257.

Mamun, Q., 2014. A coverage-based scheduling algorithm for WSNs. Int. J. Wirel. Inf. Netw. 21, 48–57.

Min, J.-K., Ng, R.T., Shim, K., 2015. Aggregate query processing in the presence of duplicates in wireless sensor networks. Inform. Sci. 297, 1–20.

Mittal, R., Agrawal, S., Das, M.L., 2016. Secure node localization in clustered sensor networks with effective key revocation. In: Emerging Innovations in Wireless Networks and Broadband Technologies. p. 12.

Osborne, M.J., 2004. An Introduction To Game Theory. Oxford University Press, New York.

Shih, E., Cho, S.-H., Ickes, N., Min, R., Sinha, A., Wang, A., Chandrakasan, A., 2001. Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking. ACM, pp. 272–287.

Stanoev, A., Filiposka, S., In, V., Kocarev, L., 2016. Cooperative method for wireless sensor network localization. Ad Hoc Netw. 40, 61–72.

Tezcan, N., Wang, W., 2007. Effective coverage and connectivity preserving in wireless sensor networks. In: 2007 IEEE Wireless Communications and Networking Conference. IEEE, pp. 3388–3393.

Wang, B., Fu, C., Lim, H.B., 2009. Layered diffusion-based coverage control in wireless sensor networks. Comput. Netw. 53, 1114–1124.

Wang, N., Zhang, N., Wang, M., 2006. Wireless sensors in agriculture and food industry—recent development and future perspective. Comput. Electron. Agric. 50, 1–14.

Wu, Y., Cardei, M., 2016. Distributed algorithms for barrier coverage via sensor rotation in wireless sensor networks. J. Comb. Optim. 1–22.

Yang, Q., He, S., Li, J., Chen, J., Sun, Y., 2015. Energy-efficient probabilistic area coverage in wireless sensor networks. IEEE Trans. Veh. Technol. 64, 367–377.

Zhang, H., Hou, J.C., 2005. Maintaining sensing coverage and connectivity in large sensor networks. Ad Hoc Sensor Wirel. Netw. 1, 89–124.

Zhu, C., Zheng, C., Shu, L., Han, G., 2012. A survey on coverage and connectivity issues in wireless sensor networks. J. Netw. Comput. Appl. 35, 619–632.