Journal of Network and Computer Applications 36 (2013) 262–273

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

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

Sensing task assignment via sensor selection for maximum target coverage in WSNs

Marjan Naderan, Mehdi Dehghan *, Hossein Pedram

Computer Engineering and Information Technology Department, Amirkabir University of Technology, 424 Hafez Avenue, Tehran, Iran

ARTICLE INFO

Article history:
Received 30 September 2011
Received in revised form
23 April 2012
Accepted 7 June 2012
Available online 16 June 2012

Keywords:
WSN
Target coverage
Task assignment
Sensing range
Sensor selection
Dual decomposition

ABSTRACT

In this paper, we consider the problem of assigning the sensing task to cover maximum number of targets while minimizing the energy consumption of the sensing operation. To this end, we define the sensing task as an optimization problem of adjusting the sensing range parameter jointly with selection of nodes in a target coverage mission. We derive an energy consumption model for the sensing operation and propose a distributed greedy-based heuristic. Each node extracts a priority value based on its utility function, which is related to the distances of targets from that node. Nodes with less priority reduce their sensing range before their neighbors and optimal adjustment of sensing range of active nodes is done via a dual-based algorithm. We further extend the algorithm for scenarios with dynamic movements of targets and for localization of possible targets between neighboring nodes. Comparison of simulation results with three other methods, shows an average reduction of 30% and a maximum of 38% in the sensing energy consumption by reducing the overlaps at an average of 56%. As a result of adjusting the sensing ranges and reduced overlaps, energy consumption is distributed more uniformly in the network, which consequences in increasing the network lifetime by 26%.

1. Introduction

Wireless Sensor Networks (WSNs) consist of small-sized low cost sensor nodes with limited resources of computation, energy and wireless communications. Researchers are mostly interested in these networks for applications including long time monitoring of environments, event detection and mobile target tracking missions. Target tracking via WSNs encompasses challenging problems such as, target (point) coverage (Wang, 2010; Karl and Willig, 2005; Ghosh and Das, 2008), tracking protocols (Naderan et al., in press-a), target localization (Wang et al., 2008, 2004; Nanmaran et al., 2009), and coverage scheduling (Esnaashari and Meybodi, 2010).

From the application layer point of view, a WSN's mission (or application) can be represented as a set of tasks, in which each node runs a specific task (Frank and Romer, 2005; Nakamura et al., 2009). Common tasks in WSN's applications may be mentioned as sensing, processing, acting and communicating. In general, an application may allow assignment of more than one task on a single sensor or it may limit the number of nodes which execute the same task (Pathak and Prasanna, 2010). Abstract definitions of the task assignment problem in WSNs have used Directed Acyclic Graph (DAG) for representation, in which dependencies of tasks to each other are showed by directed edges connecting the task vertices (Pathak and Prasanna, 2010; Xie and Qin, 2008; Tian and Ekici, 2007).

In target coverage missions, the sensing range parameter is used to characterize the sensing capability of a sensor node. Different sensor types have different sensing ranges (Wang, 2010, Section 2.2.2), while more developed and complicated sensors apply different sensing ranges for one specific sensor. Examples of such adjustable sensing range nodes are available commercially as in Photoelectric Sensors, E2K-C data sheet and have been studied in some recent coverage-related researches as in Wu and Yang (2004), Wang and Medidi (2007), Boukerche and Fei (2007), Cardei et al. (2006) and Dhawan et al. (2006). In general, the sensing parts of a sensor node must be designed in a power efficient fashion, due to the fact that continuous sensor signal processing must be provided, especially in coverage related missions, while on the other hand data communication is periodic and triggered by events (Wang and Medidi, 2007; Asada et al., 1998). In particular, some recent studies have showed that increasing the sensing levels, increases network lifetime significantly and with adjustable sensing ranges the energy consumption related to the sensing operation is optimized (Wang and Medidi, 2007; Cardei et al., 2006).

In this paper, we firstly investigate energy consumption models related to the sensing operation in WSNs and derive an energy consumption model for the sensing operation in target

* Corresponding author. Postal address: Mobile Ad hoc and Wireless Sensor Networks Lab, Department of Computer Engineering and Information Technology, Amirkabir University of Technology, PO Box 15875-4413, 424 Hafez Avenue, Tehran, Iran. Tel.: +98 2164542749; fax: +98 2166495521.
E-mail addresses: naderan@aut.ac.ir (M. Naderan), dehghan@aut.ac.ir (M. Dehghan), pedram@aut.ac.ir (H. Pedram).
URL: http://ceit.aut.ac.ir/~dehghan (M. Dehghan).

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jnca.2012.06.003

coverage (and tracking) applications. We have assumed the sensing range parameter as a continuous parameter since, energy changes continuously in contrast to the discrete assumption in Cardei et al. (2006). Our energy consumption model includes three affecting parameters: (1) the distance between a sensor and target with a quadratic relation, (2) the sampling rate and (3) the number of covered targets.

Secondly, we define the sensing task assignment as an optimization problem by adjusting the sensing range parameter jointly with the selection of sensor nodes; which is based on the definition introduced in Rowaihy et al. (2008) and Misra et al. (2009). According to this model, nodes which are assigned the sensing task have to adjust their sensing ranges to cover maximum number of targets while minimizing the related energy consumption. Nodes that cannot cover any target or their nearby targets are covered by neighbor nodes are not assigned the task and they are turned off. We define the utility of a node as a function of the distances of all targets to that node. The resulted optimization problem is Mixed Integer Programming (MIP) and it is NP-hard, thus, we turn into heuristic algorithms to reach near optimal results. We present a distributed heuristic, which is a combination of a heuristic together with a dual-based method. We extend the algorithm to include scenarios with dynamic movements of targets with an additional change for localization of some targets by three sensors. Results of our contribution in this study compared with Cardei et al. (2006) reveal:

- Optimized adjustment of sensing ranges by using the dual-based method.
- Decreased conflicts of packet transmissions by setting the priority timers according to the utility function.
- Reduced overlaps of sensing ranges as a consequence of decreased number of conflicts.
- Extension of network lifetime by uniformly distributing the energy consumption of nodes throughout the network.

The rest of this paper is organized as follows: in Section 2 we review some of the recent related works. In Section 3, we define our assumptions and problem formulation. In Section 4, we present the distributed solutions. In Section 5, we evaluate our algorithm by simulation experiments with discussion on the results. Finally, in Section 6, we conclude the paper.

2. Related work

In this section, we have investigated related literature under the following two categories.

2.1. Energy consumption model of sensing

Obtaining a unique or a general model for energy consumption of sensing is nearly impossible due to the wide variety of sensor types and different parameters affecting energy consumption. Reaching an approximate expression in this context requires determining the application with specifics of the sensor type (Karl and Willig, 2005). Regardless of this, some directions can be found in the literature on this concept. For instance, in Wang and Medidi (2007) and Cardei et al. (2006), the authors have used linear ($e_s=\theta(r_s)$) and quadratic ($e_s=\theta(r_s^2)$) models, with r_s as the sensing range of node s, for their simulation experiments without delving into details.

In general, the energy consumption model of sensing is different from that of communication range (Miorandi et al., 2008). Active sensors (e.g., radar and sonar), in contrast to passive ones, have a larger amount of energy consumption since sensing relies on wave propagation laws (Miorandi et al., 2008). According to examples of sensors with adjustable sensing ranges in Photoelectric Sensors and E2K-C data sheet, their operation resembles active sensors; hence, the energy consumption of the sensing operation is considerable. The most important parts of a sensor, common in most sensor types that consume energy are as (Karl and Willig, 2005; Asada et al., 1998; Raghunathan et al., 2002): (1) Signal sampling, (2) conversion of physical signals to electrical ones, (3) analog to digital conversion, and (4) spectrum analyzer operating on A/D outputs.

Conversion of physical signals to electrical ones and its quality (power, noise) depend on the received signal power, which itself depends on the distance between signal source and the sensor (Karl and Willig, 2005, Section 13.2.1). This conversion also affects the A/D convertor operation. Hence, we can deduce that the energy consumptions of parts 2 and 3 are dependent to distance. One of the relations, under the name general sensing model is as (Wang, 2010; Karl and Willig, 2005; Ghosh and Das, 2008):$\propto d^{-\beta}(s,z)$, where $\propto$ is a constant, β is the attenuation exponent and $d(s,z)$ is the Euclidean distance. In fact, the quality of the received signal attenuates with power β of distance. We assume $\beta=2$ which is also consistent with the assumptions in Wang and Medidi (2007) and Cardei et al. (2006).

For part 1, increasing the sampling rate of a sensor not only increases the energy consumption related to the sensing part, but also increases the computation and communication energies, as well. In our assumptions in Section 3, we have assumed that all targets within the same sensing range of a node are sampled with an equal rate, independent of their distance to that node.

For part 4, the more the number of events detected at the analyzer, the more is the energy consumption (Gu et al., 2009). Hence, with our target tracking application the more the number of targets detected by a sensor, the higher is its energy consumption. We have formulated these assumptions in Section 3 to derive an expression for the energy consumption related to the sensing operation.

2.2. Task assignment and sensor selection

Primary works on the joint optimization of a continuous parameter with selection variables via the NUM framework are presented in Rowaihy et al. (2008) and Misra et al. (2009). In Rowaihy et al. (2008), the sensors are allocated to a task based on the combination of sensor attributes and task requirements by adjusting the rate parameter for selected sensors. Misra et al. (2009), introduced the joint optimization problem of sensor selection with maximization of total utility which is dependent on the rate allocation problem.

In addition, the problems of task assignment and sensor selection for WSNs, apart from each other, are widely studied by the research community. For instance, Edalat et al. (2009) solve the task assignment problem with a price-based solution. Nakamura et al. (2009) assign the routing roles to nodes based on a distributed heuristic to the minimal Steiner tree. Pathak and Prasanna (2010), Xie and Qin (2008) and Tian and Ekici (2007) used the DAG mode to represent the task allocation problem. While in Xie and Qin (2008) and Tian and Ekici (2007), the authors mainly focus on task scheduling, Pathak and Prasanna (2010) aim at minimizing the communication and computation costs for task assignment. On the other hand, the problem of sensor selection has been also investigated in Rowaihy et al. (2007) and for many target tracking applications as in Wang et al. (2004), Isler and Bajcsy (2006) and Liu and Cao (2009). Some other application or problems joined with sensor selection include: utility-based selection (Bian et al., 2006), node localization (Kaplan, 2006), formulated in an information theoretic

framework (Chu et al., 2002), and via convex optimization (Joshi and Boyd, 2009).

Overall, we claim that our work is different from Rowaihy et al. (2008) in terms of problem formulation and the sensing range parameter which we aim to optimize, and it is based on the definition of task allocation in Misra et al. (2009). It differs from Pathak and Prasanna (2010) in the sense that we define the sensing task as an optimization problem, while Pathak and Prasanna assumed the abstract definition of tasks by using the binary assignment matrix representation. On the other hand, our work mostly resembles Cardei et al.'s (2006)study, as the authors also seek near optimal values of the sensing ranges in Cardei et al. (2006). While the optimization objective in Cardei et al. (2006) is maximizing the cover sets and ours is the number of covered targets, we have implemented the distributed algorithm in Cardei et al. (2006) for comparison. Moreover, we have implemented a genetic algorithm for comparison with our method since genetic algorithms, as an example of soft computing methods likewise to Esnaashari and Meybodi (2010), attempt to find near optimal solutions of optimization problems.

3. Assumptions and problem formulation

Our target tracking network consists of a set of sensor nodes S ($|S|=N$), and a set of targets, Z ($|Z|=M$). We use the selection variable for sensor nodes x_s to differentiate between active and sleep nodes, i.e. $x_s=1$ if s is active and $x_s=0$ otherwise. In other words, x_s represents the assignment of the sensing task to node s. Each sensor s, with sensing range $r_s>0$, covers a subset of targets, $Targ(s)$ where $|Targ(s)| \leq M$. $NBR(s)$ is the set of neighbor nodes of node s. Table 1 briefly shows our notation explained in this section.

We aim to find the optimal sensing range r_s of sensor nodes in a network with some targets. It is assumed that a sensor s is able to dynamically adjust its sensing range r_s by any arbitrary amount (as long as $r_s>0$), i.e., each sensor's sensing range is a continuous variable and elastic (Naderan et al., in press-b).

We divide the total duration of the target tracking network into time intervals T_η, $\eta=1,\ldots, NL$, in which targets are assumed to be static in these intervals and NL is the number of time intervals. Each interval T_η consists of a number of iterations $t_{\eta\sigma}$, $\sigma=1,\ldots, n$, such that the network converges to its optimal values. Each sensor node s has a finite supply of remaining energy, $e_s^{re}((T_\eta)$ at the beginning of time interval T_η.

We represent the consumed energy during the sensing operation at node s as a function of r_s as $e_s^{cns}(r_s)$. to obtain a relation for energy consumption of the sensing operation we consider that by increasing r_s, sensor s can cover targets with farther distances, i.e., $d(s,z)\propto r_s$. On the other hand, the A/D convertor part of the sensor consumes more energy to amplify received signals with more r_s. According to this fact and the discussion on Section 2.1 about the attenuation of the quality of the received signal, we have (Wang and Medidi, 2007; Cardei et al., 2006):

$$e_s^{cns}(r_s)\propto \begin{cases} A_s r_s^2 & \text{if } r_s^{\min}\leq r_s\leq r_s^{\max} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Another assumption of our model was its dependence on sampling rate and the number of targets detected. Suppose that the required energy for conversion of one sample signal of one target is e_{11}, and the sampling rate for all targets covered by sensor s with sensing range r_s is n_s^{smp}. We denote sampling rate by n since, it can be defined as the number of samples in one time unit (second). We can write the energy consumption of node s related to the sampling rate and number of covered targets as:

$$e_s^{cns}(r_s)\propto e_{11} n_s^{smp} |\mathrm{Targ}(s)| \tag{2}$$

Table 1
Parameters and decision variables used in our model.

Parameter/ variable	Definition
S	The set of sensor nodes
N	Cardinality of S or the number of sensor nodes
s, i	Indices for sensor nodes, as either $s\in S$ or $s=1,\ldots,N$
Z	The set of targets
M	Cardinality of set Z or the number of targets
j, k	Indices for targets, i.e., $j=1,\ldots,M$
$Targ(s)$	The set of targets covered by node s
$NBR(s)$	The set of neighbors of node s
$d(s,j)$	Euclidean distance between sensor s and target j
x_s	Selection variable for sensor node s
r_s	Sensing range of sensor s, $r_s^{min}\leq r_s\leq r_s^{max}$
T_η	A time interval with static targets, $\eta=1,\ldots,NL$
$t_{\eta\sigma}$	σth iteration of time intervalη, $\sigma=1,\ldots,$ n
e_s^{rem}	Remaining energy of node s
e_s^{cns}	Energy consumption of the sensing operation for node s
A_s	Constant relating the energy consumption with r_s
e_{11}	Required energy for conversion of one sample signal for one target
n_s^{smp}	Sampling rate for all targets covered by sensor s
n_{typ}^{smp}	Typical sampling rate that a sensor requires to detect the target
U_{sj} (.)	Utility of node s for covering target j
μ_s(.)	Energy price variable (variable according to iterations)
γ_s(.)	Overlapping price variable (variable according to iterations)
α(t)	Step size variable for $\mu_s(t)$
β(t)	Step size variable for $\gamma_s(t)$
θ	Constant for the degree of overlaps, $0\leq\theta\leq 1$

We also assumed that all targets within the sensing range r_s of node s are sampled with an equal rate, independent of their distance to sensor node s. Hence, we have:

$$n_s^{smp}=\left\lceil \frac{n_{typ}^{smp}}{r_s} \right\rceil \tag{3}$$

where n_{typ}^{smp} is the typical sampling rate that a sensor requires to detect the target. By combining (1) and (2) and replacing n_s^{smp} in (2) with (3), we have:

$$e_s^{cns}(r_s)= A_s r_s^2 e_{11}\frac{n_{typ}^{smp}}{r_s}|\mathrm{Targ}(s)| = A_s e_{11} n_{typ}^{smp}|\mathrm{Targ}(s)| r_s \tag{4}$$

where we have omitted the $\lceil\ \rceil$ notation since, $e_s^{cns}(r_s)$ can be any fractional value.

Since, the consumed energy of a node s must be less than its remaining energy, we have:

$$e_s^{cns}(r_s)\leq e_s^{rem}\Rightarrow A_s e_{11} n_{typ}^{smp}|\mathrm{Targ}(s)| r_s\leq e_s^{rem} \tag{5}$$

which we have replaced $e_s^{cns}(r_s)$ with the relation in (4). Combining (5) with the assignment variable x_s, (the selection variable for node s) results:

$$A_s e_{11} n_{typ}^{smp}|\mathrm{Targ}(s)| r_s\leq x_s e_s^{rem} \tag{6}$$

If node s is not active, $x_s=0$ and it consumes no energy, $e_s^{cns}(r_s)=0$; thus, $e_s^{cns}(r_s)\leq 0$. If node s is active, $x_s=1$ and it has a sensing range r_s; hence, $e_s^{cns}(r_s)\leq e_s^{rem}$.

Moreover, we define the utility of node s for covering target j at distance $d(s,j)$ by the following function:

$$U_{sj}(r_s)=\begin{cases} 1-\frac{d(s,j)}{r_r} & \text{if } d(s,j)\leq r_s \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

which is non-linear in terms of r_s. Therefore, for each target j, if $\sum_{s=1}^{N} x_s U_{sj}(.)>0$ then target j is covered by at least one sensor node. Accordingly, we require:

$$r_s^{\min}\leq r_s\leq r_s^{\max},\quad r_s^{\min}>0 \tag{8}$$

which turns into:

$$r_s^{\min} x_s \le r_s \le r_s^{\max} x_s \tag{9}$$

when combined with the selection variable x_s.

We desire to maximize the total number of covered targets by nodes; thus, we can write the NUM-CS problem (NUM for Coverage and Selection) as in (10):

$$\text{NUM}-\text{CS}: \tag{10}$$

$$\text{Maximize} \quad \sum_{s=1}^{N}\sum_{j=1}^{M} \|x_s U_{sj}(r_s)\|_0 \tag{10-1}$$

$$s.t.: \quad U_{sj}(r_s) = \begin{cases} 1-\frac{d(s,j)}{r_r} & \text{if } d(s,j) \le r_s \\ 0 & \text{otherwise} \end{cases} \tag{10-2}$$

$$A_s e_{1,1}^{cns} n_{typ}^{smp} \| \sum_{j=1}^{M} x_s U_{sj}(r_s) \|_0 r_s \le x_s e_s^{rem} \quad \text{for all } s=1,\cdots,N \tag{10-3}$$

$$x_s r_s^{\min} \le r_s \le x_s r_s^{\max} \quad \text{for all } s=1,\ldots,N \tag{10-4}$$

$$x_s \in 0,1, r_s^{\min} > 0 \tag{10-5}$$

such that $\|.\|_0$ in the objective function in (10-1) is the l_0-norm which is the number of non-zero elements of a vector. Constraint (10-3) is the same as (6) with the difference of replacing $|Targ(s)|$ with norm-0. NUM-CS is solved in each time interval T_η which is supposed to have static targets. Furthermore, NUM-CS is MIP and it is NP-hard (Pathak and Prasanna, 2010; Rowaihy et al., 2008), hence, we turn into heuristics in the next section.

4. Distributed algorithms for NUM-CS

In this section, we first provide a dual-heuristic distributed algorithm, namely Dual-Heuristic based Distributed algorithm for NUM-CS or DHD-CS, as presented in Algorithm 1. We secondly extend DHD-CS, to support movements of targets in consecutive time intervals T_η. In addition, with a slight modification on DHD-CS we provide more accuracy for targets locations by using three sensors for some of the targets.

4.1. The basic algorithm

In DHD-CS, each node detects its neighbors in a prior phase by sending a special INIT message and preserves their information in a table named neighbor table. The information includes the ID of its neighbors in the first stage, and contains a list for the target IDs and a field for the sensing range of each neighbor during the next phases.

The greedy-based heuristic algorithm starts by setting the sensing range of each node to its maximum value. If node s detects no targets with its $r_s^{\max}$, it is turned off and it informs its neighbors about its state. Otherwise, the node sets a priority timer based on the $\sum U_{sj}(.)$ parameter for $j \epsilon \text{Targ}(s)$. The priority value is set such that higher priority is given to nodes with fewer numbers of covered targets. This is due to the fact that we aim to minimize the energy consumption of sensing for active nodes.

When the timer expires, the node checks to find out whether any of its targets are covered by neighbor nodes. If so, it removes such target from its list and lets the neighbor node to cover it. After checking the list of all targets in its sensing range, it goes to sleep if there is not any target left. Otherwise, the node sets the d_{sj} parameter to the distance of the farthest target not covered by any neighbor and uses the Modified DDA-COV algorithm, Algorithm 2, to reach the optimal value for the sensing range parameter. The modified DDA-COV algorithm is described next. After reaching the optimal value for r_s, node s sends this value and the list of covered targets to its neighbors by encapsulating this information in a DHDPKT packet.

On the other hand, when a node receives a DHDPKT message from one of its neighbors, it updates the corresponding entry of that node which contains the list of IDs of covered targets.

The Modified DDA-COV algorithm is very similar to the DDA-COV algorithm which we designed for each node to reach the optimal value of the sensing range in Naderan et al. (in press-b). We have provided the mathematical approach based on the dual problem to reach the Modified DDA-COV algorithm in Appendix A. The dual decomposition method has the power of finding the optimal values, according to the constraints in an optimization problem as far as the whole problem is convex. The optimization problem may include more complicated constraints as long as they remain convex.

We continue with our previous problem in Naderan et al. (in press-b) with two small modifications based on the d_{sj} parameter and the utility function. In fact, the sensing ranges of two neighbor nodes which were constrained by the Euclidean distance between them in Naderan et al. (in press-b), is now restricted by the distance of the farthest target. Furthermore, the logarithmic utility function of Naderan et al. (in press-b) is now changed to (7) to be consistent with this study. The resulting distributed iterative algorithm is presented in Algorithm 2.

Algorithm 1, DHD-CS: Dual-Heuristic based Distributed algorithm for NUM-CS, for each node *s*

Initialization:

Collect IDs of neighbor nodes by sending INIT packets
Set $r_s = r_s^{max}$
Determine number of covered targets by this r_s, $Targ(s)$
If ($Targ(s)=\emptyset$) then
  Set $x_s=0$
  Send sleep state and DHDPKT to neighbors
  Break
Send $Targ(s)$ to neighbors in a DHDPKT packet
Each node s sets a timer according to its $\sum_j U_{sj}(.)$
Compute e_s^{cns} according to (4)
Update remaining energy (related to the sensing operation) as
$e_s^{rem} = e_s^{rem} - e_s^{cns}$

Main procedure:

1. Upon expiring the timer:
  For all $j \epsilon Targ(s)$
    For all $i \epsilon NBR(s)$
      If ($j \epsilon Targ(i)$) then
        $Targ(s) = Targ(s) - j$

  If ($Targ(s)=\emptyset$) then
    Set $x_s=0$
    Send sleep state and DHDPKT to neighbors
    Break

  Set $d_{sj} = \max d(s,j)$ for all $j \epsilon Targ(s)$
  Run Modified DDA-COV algorithm to obtain optimal r_s
  Send $Targ(s)$ and new r_s to neighbors in a DHDPKT
  If ($r_s \neq r_s^{max}$) then
    Compute e_s^{cns} according to (4)
    Update remaining energy (related to the sensing operation) as
    $e_s^{rem} = e_s^{rem} - e_s^{cns}$

  If ($e_s^{rem} \le \varepsilon$)

Set $x_s=0$
Send sleep state and DHDPKT to neighbors
Break

2. Upon receiving a DHDPKT from any neighbor node:
Update the corresponding entry in the neighbor table containing the list of covered targets of neighbor nodes.

Algorithm 2: Modified DDA-COV for each node *s*

Initialization: set $r_s(0)=r_s^{min}$, $d_{sj}=\max d(s,j)$ for all $j\epsilon Targ(s)$, and the values of $\mu_s(0)$, $\gamma_s(0)$, $\alpha(0)$ and $\beta(0)$ to arbitrary positive values.

While ($|r_s(t+1)\text{-}r_s(t)| > 0.001$) and ($t < iterationLimit$) do

1) Node s updates its energy and overlapping prices according to:

$$\mu_s(t+1)=[\mu_s(t)-\alpha(t)(e_s^{rem}-A_s e_{1,1}^{cns} n_{typ}^{smp}|Targ(s)|r_s(t))]^+$$

$$\gamma_s(t+1)=[\gamma_s(t)\text{ - }\beta(t)((1+\theta)d_{sj}-r_s(t))]^+$$

2) Node s locally finds its new sensing range $r_s(t+1)$ for the next iteration by:

$$L'=\sqrt{\frac{d_{sj}}{\left(\mu_s(t)A_s e_{1,1}^{cns} n_{typ}^{smp}|\mathrm{Targ}(s)|+\gamma_s(t)\right)}}$$

$$r_s(t+1)=\begin{cases} L' & \text{if } r_s^{\min}\le L'\le r_s^{\max}\\ r_s^{\min} & \text{if } r_s^{\min}>L'\\ r_s^{\max} & \text{if } L'>r_s^{\max}\end{cases}$$

3) $\alpha(t+1)=\beta(t+1)=\kappa\ \alpha(t)$

4) $t=t+1$

4.2. Extension for dynamic movements of targets

To extend the DHD-CS algorithm for movements of targets in consecutive time intervals, we have to consider two cases:

- A sensor node detects a target has moved out of its sensing range.
- A sensor node detects a target has entered into its sensing range.

In the first case, suppose node s detects target j has moved out of its sensing range. If target j was the farthest covered target, s informs neighbor nodes to find out whether target j is in their sensing region or not by sending the message SEARCH_TARG. According to our assumption of dividing the total network lifetime into time intervals with static targets, it is assumed that a target cannot move distantly between two consecutive time intervals. Hence, when target j moves out of the sensing region of node s, it certainly moves to the sensing region of a neighbor node.

If any neighbor node detects target j without increasing its sensing range, it sends a TARG_FIND message back to node s. Node s also sends TARG_FIND to its neighbors to inform them that target j is found and reduces its r_s to $r_s=\max d(s, j)$. If node s does not receive any replies from its neighbors (in response to the SEARCH_TARG message), it concludes it is better to increase the sensing range of all nodes to $r_s^{\max}$ and run the DHD-CS algorithm. Hence, it sends INC_RS message and DHD-CS algorithm is executed.

The second case only happens when an additional target from outside enters the monitoring region. If a target remains inside the monitoring region and a sensor node detects it, this target has certainly moved out of the sensing range of a neighbor node. Therefore, a SEARCH_TARG message for this target was received previously. Hence, for the second case, node s has detected a new target j in its sensing region and it has not received a SEARCH_TARG message before. Thus, node s sends the list of its targets to neighbors in a DHDPKT. On the other hand, if it has received a SEARCH_TARG before, it replies with a TARG_FIND message and sends the list of its targets to its neighbors. The pseudo code of the complete algorithm for Dynamic DHD-CS and a flowchart of it are presented in Algorithm 3 and Fig. 1, respectively. The three cases mentioned in the bottom of Fig. 1 are used in Section 4.4 for evaluation of packet exchanges.

Algorithm 3, Dynamic DHD-CS: DHD-CS for dynamic target movements,for each node s

Initialization:
Run DHD-CS at least one time so that each node reaches a sensing range.

Main procedure:

1. If (node s detects target j has moved out of its sensing region) then
Send SEARCH_TARG to neighbors.
Set timer TIMER1

2. If (node s detects target j has entered its sensing region) then
If (this node has not received any SEARCH_TARG) then
Send $Targ(s)$ to neighbor nodes

Else
Send TARG_FIND to neighbors.
Send $Targ(s)$ to neighbors.

3. Upon expiring TIMER1:
If (this node has not received any TARG_FIND) then
Send INC_RS to neighbors.
Run DHD-CS algorithm.

4. Upon receiving a SEARCH_TARG message:
If (this node detects a new target) then
Sends TARG_FIND to the sender of SEARCH_TARG message

5. Upon receiving a TARG_FIND message:
If (this node has sent a SEARCH_TARG) then
Update the entry corresponding to the sender of TARG_FIND message in the neighbor table.
Send TARG_FIND to neighbors.
Update $maxDsj$: $maxDsj=\max d(s, k)$ for all $k\epsilon Targ(s)\text{-}j$.
Run Modified NUM-COV to find optimal r_s.

6. Upon receiving a INC_RS message:
Run DHD-CS algorithm.

4.3. Modification for target localization

The basic DHD-CS algorithm reduces the sensing ranges of nodes to not cover common targets due to energy considerations. In spite that with the factor θ in the Modified DDA-COV algorithm (which is also existent in the DDA-COV algorithm), a node controls the degree of overlap between its own sensing ranges and that of its neighbors, another improvement can be made to the algorithm to localize possible targets between three or more sensor nodes.

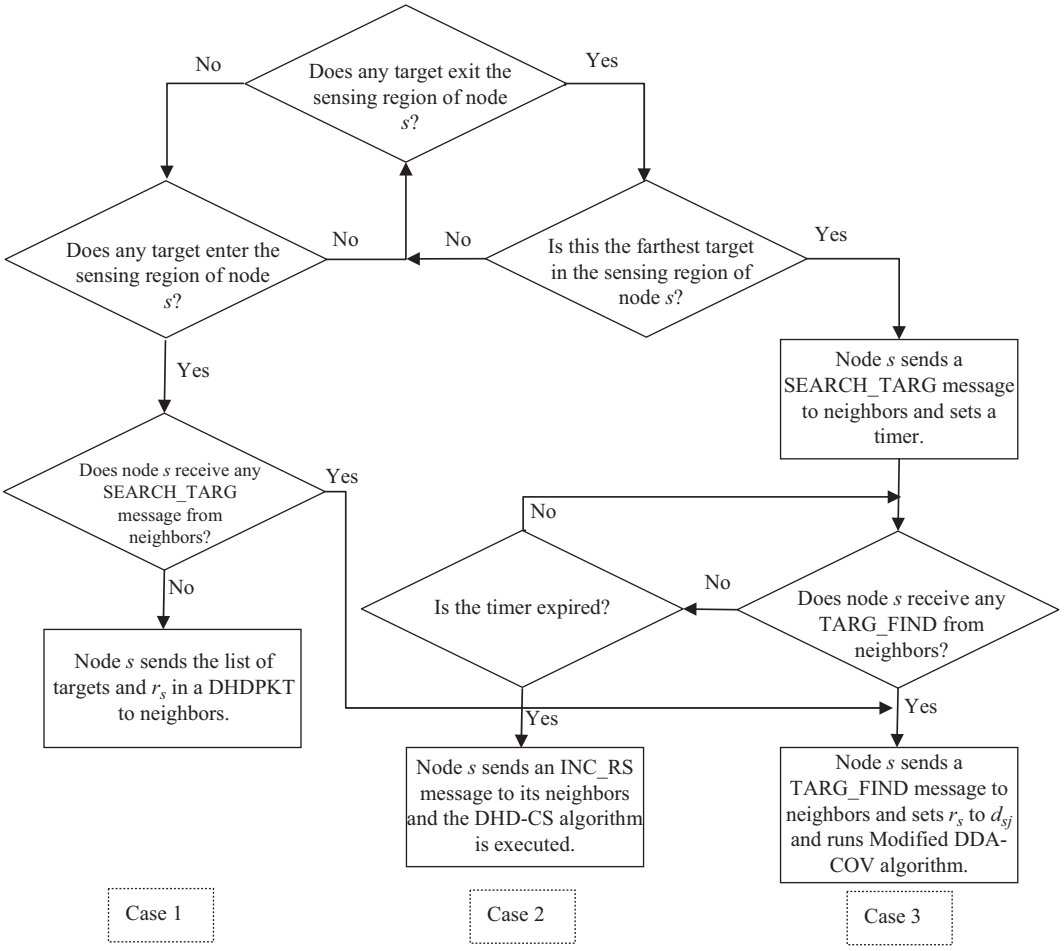

Fig. 1. Flowchart for algorithm Dynamic DHD-CS.

To this end, we make a modification to the basic DHD-CS algorithm when each node is deciding to reduce its sensing range. By this modification, if node s detects that a target j is covered by more than three neighboring nodes, it can determine its location according to the Trilateration algorithm. Otherwise, in case of covering the target by one or two neighbors, it removes that target from its list which is similar to the basic DHC-CS algorithm. With this improvement, mostly common targets can be localized by three sensors, since they are more likely to be covered by overlaps of sensing ranges of neighbor nodes. The changes in the pseudo code of DHD-CS for this modification, namely DHD-CS-LOC, are presented in Algorithm 4.

4.4. Analytical evaluations for overheads of algorithms

In this section, we analyze the overhead of algorithms by evaluations of their packet exchanges and time. The packet overhead of DHD-CS algorithm is similar to Cardei et al.'s (2006) distributed algorithm. At worst case, each node sends a DHDPKT to its neighbors with a single MAC layer broadcast upon time r expiration. Therefore, packet overhead of DHD-CS algorithm is $O(N)$. Its time complexity is dependent on the sum of total timer values, which may take a long time when the number of nodes is high. We have verified this result in Naderan et al. (in press-b) for 300 nodes and more.

In DHD-CS timers are set according to the utility function which is a function of distances of targets to nodes. Compared to Cardei's algorithm, this timer selection results in more distinct values. Consequently the collision of packets is reduced and on the other hand, the number of correct packet transmissions is increased. This result is verified and more discussed in Section 5.

Packet overhead of Dynamic DHD-CS algorithm is presented in Table 2. According to the cases mentioned in Fig. 1, for case 1, just one broadcast of DHDPKT to neighbors is needed. For case 2, one broadcast for SEARCH_TARGET, and one broadcast of INC_RS to neighbors are needed and after that the DHD-CS algorithm is executed. For case 3, one broadcast of SEARCH_TARGET, two broadcasts of TARG_FIND and one broadcast of the DHD-PKT are needed. It can be seen that the addition of external packets for Dynamic DHD-CS is very low.

Modification of the DHD-CS algorithm for localization imposes no packet overhead, as can be seen from Algorithm 4. Only a condition is checked to determine whether the target is covered by three or more neighbors or not.

Finally, the convergence of the Modified DDA-COV algorithm is strongly dependent to the selection of positive step sizes. This issue and the choice of step sizes in this paper are mentioned in Appendix A and in Naderan et al. (in press-b). Furthermore, the iterations loop is executed internally inside each node, and there is no need for each node to exchange the values with its neighbors at each iteration, in contrast to the DDA-COV algorithm in Naderan et al. (in press-b).

Table 2
Transmitted packets overhead in Dynamic DHD-CS.

Case	Overhead of packets
1	1 broadcast of DHDPKT to neighbors
2	1 broadcast of SEARCH_TARG to neighbors + 1 broadcast of INC_RS to neighbors + 1 run of DHD-CS
3	1 broadcast of SEARCH_TARG to neighbors + 2 broadcasts of TARG_FIND to neighbors + 1 broadcast of DHDPKT to neighbors

Algorithm 4, DHD-CS-LOC: modification to DHD-CS algorithm for localization, for each node s
Main procedure:

1. Upon expiring the timer:
2. For all $j \epsilon Targ(s)$
   Set $cnt = 0$
   For all $i \epsilon NBR(s)$
   If $j \epsilon Targ(i)$ then
   $++cnt$

   If $cnt \geq 3$
   Target j can be localized by at least 3 sensors

   If $1 \leq cnt \leq 2$
   Target j is covered by another neighbor
   $Targ(s) = Targ(s) - j$

3. If $Targ(s) = \emptyset$ then
   $x_s = 0$.
   Send sleep state and DHDPKT to neighbors.
   Break.

5. Simulation results

In this section, we define the series of simulation experiments to evaluate the performance of the basic DHD-CS algorithm. We implemented the algorithm in Castalia 3.2 simulator (The Castalia Website), which is based on OMNeT++ platform (The OMNet++ Website). The main modules in Castalia have modeled sensor nodes, wireless channel and physical processes. We used the physical and MAC layers and implemented DHD-CS, Cardei (Cardei et al., 2006), DDA-COV (Naderan et al., in press-b), and a genetic algorithm in the application layer for comparison. The main drawback of the two latter algorithms is that they do not guarantee that the optimal value(s) are found. In the DDA-COV and genetic algorithms, all nodes are active, even if there are not any targets in the maximum sensing range of some nodes. In contrast, in DHD-CS and Cardei's algorithms nodes that detect no targets within any sensing range are turned off, hence, more energy is conserved. We have used six measures to evaluate these algorithms:

- energy consumption related to the transmission of packets, i.e., send and receive;
- energy consumption related to the sensing operation;
- average number of transmitted packets per node;
- sum of logarithms of sensing ranges of all nodes;
- percentage of sleep nodes;
- total network lifetime.

Since the energy consumption of wireless transmission is much higher than that of the sensing operation, we used the average energy per node for the first measure and the sum of energies of all nodes for the second measure. Moreover, the sum of logarithms of r_s of all nodes presents a measure of the length of sensing ranges, which are going to be adjusted by the DHD-CS algorithm.

Table 3
Number of nodes with the corresponding size of the area and number of targets.

Number of nodes	Size of the network (m^2)	Number of targets
50	150×150	25
100	200×200	50
200	400×400	100
300	500×500	150
400	600×600	200
500	700×700	250
600	800×800	300
700	900×900	350
800	1000×1000	400
900	1100×1100	450
1000	1200×1200	500

5.1. Scenarios and initial values

We have used networks with random placements of nodes and targets in all scenarios. Parameters of the Modified DDA-COV algorithm are chosen according to Naderan et al. (in press-b): values of the step sizes, $\alpha(t)$ and $\beta(t)$, are $\alpha(t+1) = \beta(t+1) = \kappa\alpha(t)$, in which $0 < \kappa < 1$ and we used $\kappa = 0.7$. Thirty iterations are needed for convergence of the Modified DDA-COV.

The initial energy of a node, which is spent during the transmission operations (send/receive), is set to 18,720 Jules, the typical energy of two AA batteries used in Castalia 3.2. We have added the energy resource of the sensing operation as a separate part and each node has an initial value of 100 Jules for the sensing operation. Maximum sensing range for each node is set to 30 m and for Cardei's algorithm 6 levels between 1 and 30 m are chosen arbitrarily as {1.1, 7.5, 15, 20, 25, 30} as advised by Cardei et al. (2006). The communication range of nodes is set to be nearly 20 m (-5 dB as the transmission power), the radio model is CC2420 (CC2420 data sheet, 2007) and the MAC layer protocol used is TMAC implemented in Castalia 3.2 (Tselishchev et al., 2010). Each point of the graphs is the average of five runs for random placement of nodes and targets, and each run of the algorithms lasts for 3600 s. We have increased the number of nodes, number of targets and the area according to Table 3. Parameters and their values for the genetic algorithm are also given in Table 4.

5.2. Results for the DHD-CS algorithm

Figure 2 shows the average energy consumption per node during the communications for DHD-CS in comparison with Cardei, DDA-COV and genetic algorithms. It can be seen that DHD-CS and Cardei's algorithms have the same energy consumption while, the DDA-COV and genetic algorithms have near each other values with an extreme difference from DHD-CS and Cardei. This is due to the iterative-based nature of the DDA-COV and genetic algorithms in which, each node sends its sensing range to its neighbors in each iteration and, a total of 30 iterations are executed in these simulations. Furthermore, the reason that DHD-CS and Cardei have nearly the same energy consumption during communication, despite DHD-CS has a few more packet exchanges, is that the energy consumption of nodes for communication is dependent on their time of activity and the

Table 4
Parameters and their value for the genetic algorithm.

Parameter	Value
Population number	20
Number of generations	30 (the same as the number of iterations)
Number of runs for each scenario	20
Distribution of the current population if it is better than the previous one	30% the best chromosome, 40% children by mutation, 40% children by cross over
Distribution of the current population if it is worse than the previous one	50% the best chromosome, 50% children by mutation

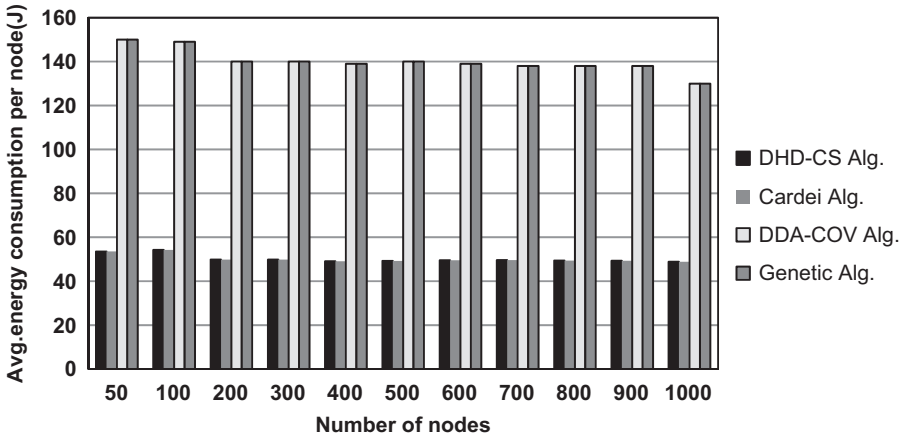

Fig. 2. Average energy consumption of communication per node.

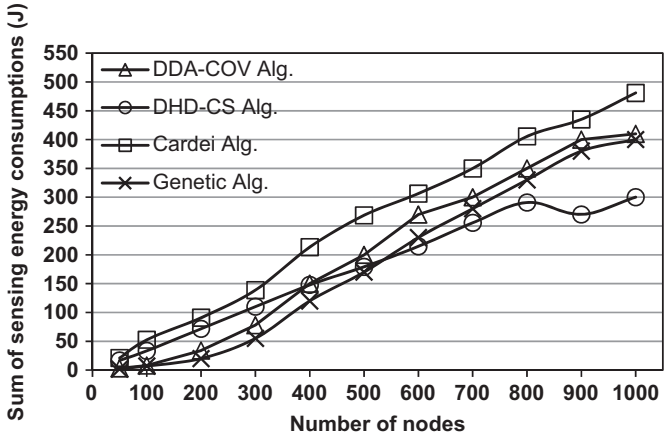

Fig. 3. Sum of energy consumptions of all nodes related to the sensing operation.

operation of the MAC protocol. Hence, in DHD-CS and Cardei's algorithms nodes have nearly the same time of activeness, while for DDA-COV and genetic algorithms this time is much more since, they need a specific number of iterations.

Figure 3 shows sum of energy consumptions of all nodes relating to the sensing operation for DHD-CS compared to the other three algorithms. It can be seen that for small number of nodes (less than 300 nodes), the algorithms have nearly the same amount of energy consumption, while for large numbers of nodes (more than 600 nodes) our DHD-CS algorithm has the least energy consumption. DDA-COV and the genetic algorithms have near each other values and their values are more than DHD-CS and less than Cardei's algorithm. The reason that these two algorithms perform better than Cardei's in Fig. 3 is that despite of their iteration-based operation, they adjust the sensing ranges of nodes better. In addition, Cardei's algorithm always consumes more energy for the sensing operation than the other three algorithms, which shows nodes have a lot of overlap in their sensing ranges (this issue is also approved by Fig. 6). In fact, in Cardei's algorithm nodes start with r_s^{max} and the priority timer is set such that nodes with more targets have higher priority. Thus, at the end of algorithm operation, nodes with more sensing ranges stay active. The maximum difference between DHD-CS and Cardei's algorithms seen in Fig. 3 is for 1000 nodes, which is 38% reduction in the energy consumption for the sensing operation and an average value of 30% reduction.

In addition to the energy consumption metric, we have measured the average number of packets transmitted per node in Fig. 4. It can be seen from this figure that DHD-CS and Cardei's algorithms have nearly the same values of packet transmissions while the DDA-COV and genetic algorithms have extremely a large value for this parameter. This is due to the different nature of the two latter algorithms, which are iterative-based algorithms and packet transmissions are needed in each iteration of the algorithm.

In fact, in DDA-COV and the genetic algorithm, each node sends its sensing information to its neighbors and consequently receives from them in every iteration. On the other hand, DHD-CS and Cardei's algorithms are based on priority timers and sensor nodes with no targets are turned off, which cause great savings in the number of transmitted packets. Moreover, DHD-CS has a few numbers of packets more than Cardei's algorithm, which we have illustrated this difference in Fig. 5.

As seen from Fig. 5, DHD-CS algorithm still has less than 10 packets. Furthermore, for large numbers of nodes (900 and 1000 nodes) the average number of transmitted packets for the two algorithms reaches nearly the same values.

Figure 6 shows sum of logarithms of sensing ranges, as a measure of values of sensing ranges, for the four algorithms. It can be seen that the DDA-COV and genetic algorithms have close values, while they stand mostly between values of DHD-CS and Cardei's algorithms. Considering this figure with Fig. 3 represents that, by using DHD-CS algorithms nodes adjust their sensing ranges with less overlaps and with average reduction of 56% compared to Cardei's values. In addition, despite we stated in Naderan et al. (in press-b) that by

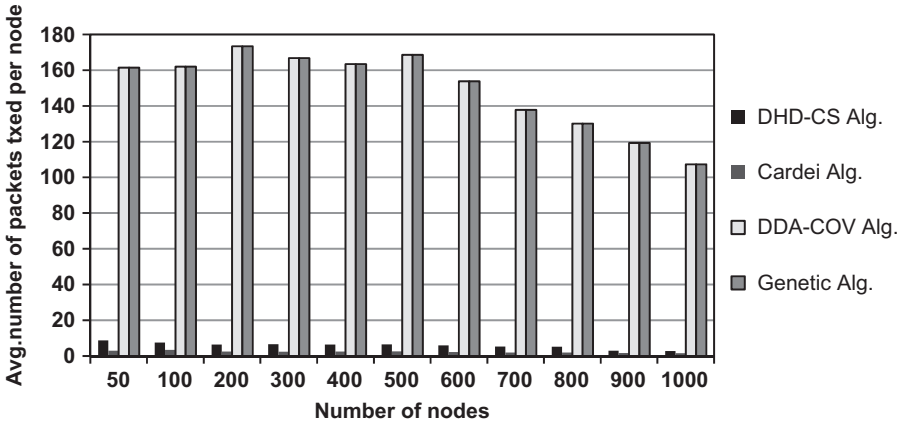

Fig. 4. Average number of packets transmitted per node.

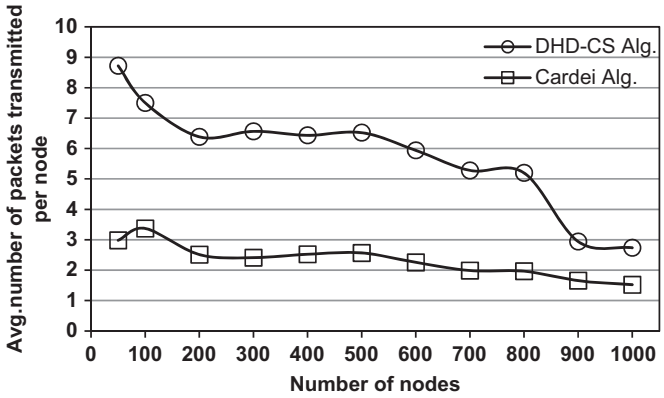

Fig. 5. Average number of packets transmitted per node for DHD-CS and Cardei's algorithms.

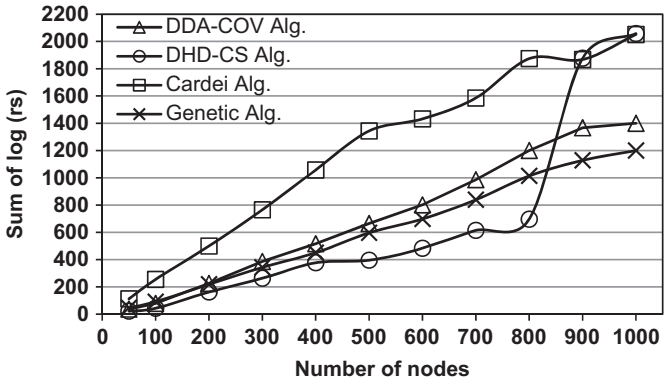

Fig. 6. Sum of $\log(r_s)$ of nodes.

DDA-COV sensor nodes adjust their sensing ranges to optimal values, the reason that DHD-CS has less values than DDA-COV in Fig. 6 is that in DHD-CS nodes with no targets are turned off, while in DDA-COV all nodes are active. The genetic algorithm also has near values to DDA-COV since it reaches near optimal values with all nodes being active. On the other hand, Cardei's algorithm operates the worst in adjusting the sensing ranges and nodes have a lot of overlaps.

Two main reasons hold for our DHD-CS algorithm to adjust the sensing ranges better than Cardei's algorithm. The first is the usage of Modified DDA-COV which adjusts the sensing ranges according to the optimal value. The second reason is the reduction of overlaps by using a different priority timer in DHD-CS. In Cardei et al. (2006), the value of priority timer for node s, W_s, is set as:

$$B_{s,\max} = \frac{T_{s,\max}}{e_{\max}} \tag{11}$$

$$W_s = \left(1 - B_{s,\max}/B_{\max}\right) \tag{12}$$

which indicates that the initial priority timer, W_s, is dependent on the number of covered targets, $T_{s,\max}$. Therefore, nodes with same values of covered targets have the same priority timers, which consequently results in transmission conflicts. This result is also mentioned by Cardei et al. (2006) and approved here by simulation experiments. On the other hand, we proposed the value of priority timer for each node s as $\sum u_{sj}$ for $j \epsilon Targ(s)$, which is dependent on distances of nodes to targets. This consequences in more distinct values for priority timers of nodes, and accordingly reduces conflicts of packet transmissions. From this discussion, and Figs. 3 and 6, it can be deduced that our DHD-CS algorithm has reduced the problem of transmission conflicts to a great extent. On the other hand, by setting higher priority for nodes with fewer targets, the heuristic results in minimizing the energy consumption of the sensing operation for active node, i.e., maximizing target coverage by minimizing the energy consumption.

Finally, Figure 7 presents the percentage of sleep nodes of our DHD-CS algorithm in comparison with Cardei's. In DDA-COV and the genetic algorithms all nodes are active, thus, they are not included in this figure. It can be seen that Cardei's algorithm has more number of sleep nodes than DHD-CS, and this parameter decreases nearly by 20% with increasing the number of nodes. This result may mislead us at the first glance to the fact that Cardei's algorithm performs better in terms of energy consumption. However, by considering Figs. 2 and 3, we conclude that energy consumption is dependent on the duration of sleep/active periods of nodes. In fact, merely sleeping nodes at the application level does not lead to the reduction of energy consumption, since nodes are periodically awakened in the MAC layer to receive any probable packets.

By considering also the previous discussion on Fig. 6 about the sensing ranges, we conclude that Cardei's algorithm sets most of the nodes to the sleep mode and increases the sensing ranges of active nodes despite having overlaps. In contrast, our DHD-CS algorithm sets a smaller number of nodes to sleep and adjusts the sensing ranges of active nodes more accurately. Overall, this results in a more uniform distribution of energy consumption of nodes throughout the network and increases the network lifetime. This conclusion is depicted in Fig. 8, which shows if the network lifetime is divided into rounds (for DHD-CS and Cardei's algorithm), the whole network lifetime is the sum of all rounds when all nodes are alive. According to Fig. 8, an average of 26%

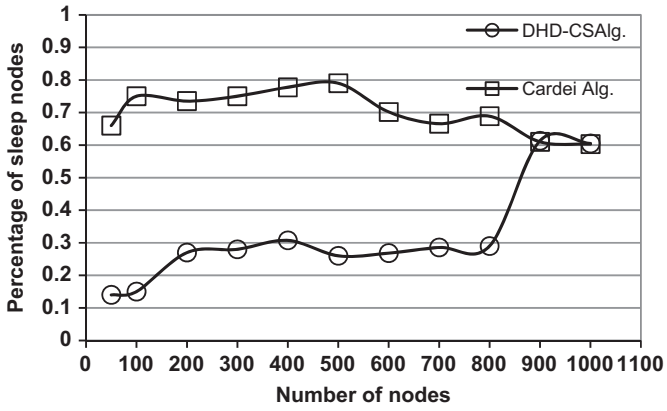

Fig. 7. Percentage of sleep nodes.

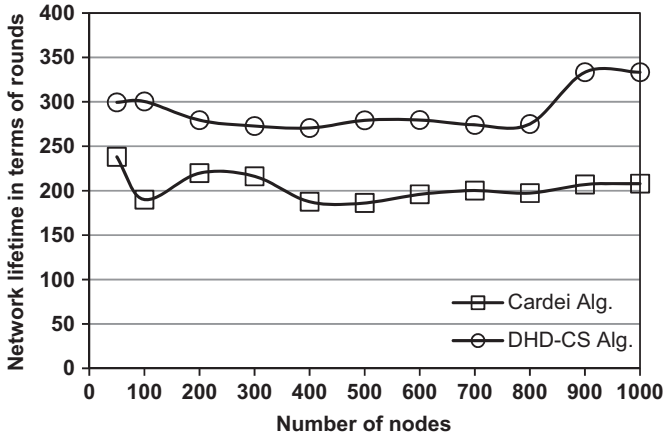

Fig. 8. Network lifetime in terms of rounds.

improvement in the total network lifetime is reached by DHD-CS in comparison with Cardei's algorithm.

6. Conclusion and future work

In this paper, we modeled a target coverage mission via the sensing task assignment problem to maximize the number of covered targets subject to energy constraints. We defined a utility function for nodes which depends on the sum of distances of targets to each node in an MIP problem and used a selection variable to turn off nodes with no covered targets. Since the energy constraint of the problem is dependent on the sensing operation, we discussed energy consumption models related to the sensing operation and proposed a relation for it according to the distance between a sensor and target, the sampling rate and the number of covered targets.

We presented a basic greedy-based heuristic algorithm, DHD-CS, according to the NP-hardness nature of the problem and derived an extension for dynamic movements of targets. With a small modification, localization for some targets by three sensor nodes is also possible. We implemented our algorithm with three other methods using Castalia framework and showed that it operates with a maximum of 38% and an average of 30% reduction in the sensing energy consumption while adjusting sensing ranges more accurately by reducing their overlaps to an average of 56%. Furthermore, by accurately adjusting the sensing ranges and turning off unnecessary nodes, DHD-CS distributes energy consumption of nodes more uniformly than other comparing algorithms and extends network lifetime by an average of 26%.

We are continuing the research on this work by implementing the dynamic extension of DHD-CS algorithm for movements of targets. An important future work for our problem is to extend the single task assignment to two and more tasks by using a binary assignment matrix representation together with optimizing other continuous parameters for every task.

Acknowledgment

This work is supported in part by Iran Telecommunications Research Center (ITRC) under grant #1042/500.

Appendix A. The modified DDA-COV algorithm

In this section, we derive the relations used in the Modified DDA-COV algorithm. The Modified NUM-CS problem similar to the NUM-COV problem of Naderan et al. (in press-b) is as:

$$\text{Modified NUM}-\text{COV}(U,A,E_{rem}(T_\eta)): \tag{13}$$

$$\text{Maximize} \quad \sum_{s=1}^{N}\sum_{j=1}^{M} U_{sj}(r_s) \tag{13-1}$$

$$s.t.: \quad e_s^{cns} = A_s e_{1,1}^{cns} n_{typ}^{smp} |\text{Targ}(s)| r_s \le e_s^{rem}(T_\eta) \quad s=1,\ldots,N \tag{13-2}$$

$$r_s \le (1+\theta) d_{sj} \quad d_{sj} = \max d(s,j) \quad \text{for all} j \in \text{Targ}(s) \tag{13-3}$$

$$r_s^{\min} \le r_s \le r_s^{\max} \tag{13-4}$$

with the utility function defined in (7), the first constraint is as (5) and the second constraint limits the r_s with the distance of the farthest target in its range. The Lagrangean of the Modified NUM-COV problem is:

$$\begin{aligned} L_{MOD-NUM-COV}(r,\mu,\gamma) = &\sum_{s=1}^{N}\sum_{j=1}^{M} U_{sj}(r_s) \\ &+ \sum_{s=1}^{N} \mu_s \left(e_s^{rem} - A_s e_{1,1}^{cns} n_{typ}^{smp} |T\text{arg}(s)| r_s \right) \\ &+ \sum_{s=1}^{N} \gamma_s ((1+\theta) d_{sj} - r_s) \end{aligned} \tag{14}$$

where μ_s and γ_s are interpreted as energy and overlapping prices, respectively. Also, compared to the γ_{si} parameters for all $i \in NBR(s)$ in Naderan et al. (in press-b), we have one parameter γ_s for each node s. We can rewrite it as:

$$\begin{aligned} L_{MOD-NUM-COV}(r,\mu,\gamma) = &\sum_{s=1}^{N} \left(\sum_{j=1}^{M} U_{sj}(r_s) - \mu_s A_s e_{1,1}^{cns} n_{typ}^{smp} |T\text{arg}(s)| r_s - \gamma_s r_s \right) \\ &+ \sum_{s=1}^{N} (e_s^{rem} \mu_s + \gamma_s (1+\theta) d_{sj}) \end{aligned} \tag{15}$$

From the above expression, the Lagrangean can be separated into many sub-problems on r_s, hence, the dual function (for each sensor s) is:

$$\begin{aligned} &D(\mu_s,\gamma_s) = \text{Maximize} \quad L_{MOD-NUM-COV}(r_s,\mu_s,\gamma_s) \\ &s.t.: \quad r_s^{\min} \le r_s \le r_s^{\max} \end{aligned} \tag{16}$$

and the master dual problem is:

$$\text{Minimize} \quad D(\mu_s,\gamma_s)$$
$$s.t.: \quad \mu_s \geq 0,\ \gamma_s \geq 0 \tag{17}$$

Since U_s is concave and the constraints in Modified NUM-COV are linear there is no duality gap and dual optimal prices, which are Lagrange multipliers, exist. Hence, we first solve (17). To obtain dual optimal prices, μ_s^* and γ_s^*, we use the gradient descent algorithm to update the Lagrange multipliers as:

$$\mu_s(t+1) = \left[\mu_s(t) - \alpha(t)\left(e_s^{rem} - A_s e_{1,1}^{cns} n_{typ}^{smp} |\text{Targ}(s)| r_s(t)\right)\right]^+ \tag{18}$$

$$\gamma_s(t+1) = \left[\gamma_s(t) - \beta(t)\left((1+\theta)d_{sj} - r_s(t)\right)\right]^+ \tag{19}$$

where $[.]^+$ denotes the projection onto the set R^+ of non-negative real numbers and $\alpha(t)$ and $\beta(t)$ are positive scalar step sizes. Now that we have obtained $\mu^*=(\mu_s^*, s=1,\ldots,N)$ and $\gamma^*=(\gamma_s^*, s=1,\ldots,N)$, the primal optimal $r^*=r(\mu^*, \gamma^*)$ can be computed by individual nodes s locally. The primal optimal r^* is obtained by considering (16). Since the Lagrangean is separable, this maximization of Lagrangean over r_s can be conducted in parallel at each source s by:

$$\text{Maximize} \quad \sum_{j=1}^{M} U_{sj}(r_s) - \mu_s A_s e_{1,1}^{cns} n_{typ}^{smp} |\text{Targ}(s)| r_s - \gamma_s r_s$$
$$s.t.: \quad r_s^{\min} \leq r_s \leq r_s^{\max} \tag{20}$$

To solve (17), each node takes the derivative of the objective function of (20) in the range $[r_s^{min}, r_s^{max}]$ and by assuming $U_{sj}(r_s)$ as (7), which results in:

$$L' = \sqrt{\frac{d_{sj}}{\left(\mu_s(t) A_s e_{1,1}^{cns} n_{typ}^{smp} |\text{Targ}(s)| + \gamma_s(t)\right)}}$$
$$r_s(t+1) = \begin{cases} L' & \text{if} \quad r_s^{\min} \leq L' \leq r_s^{\max} \\ r_s^{\min} & \text{if} \quad r_s^{\min} > L' \\ r_s^{\max} & \text{if} \quad L' > r_s^{\max} \end{cases} \tag{21}$$

It is obvious that by Eqs. (18) and (19), dual variables $\mu_s(t)$ and $\gamma_s(t)$ converge to the optimal dual solutions $\mu_s^*(t)$ and $\gamma_s^*(t)$ if the step sizes are chosen such that

$$\alpha(t) \to 0, \sum_{t=1}^{\infty} \alpha(t) = \infty, \beta(t) \to 0, \sum_{t=1}^{\infty} \beta(t) = \infty \tag{22}$$

Eq. (22) are known as non-summable diminishing step size rule (Boyd and Vandenberghe, 2004). Other choices for determining the values of step sizes may be mentioned as the constant value step size rule or another which we used in Naderan et al. (in press-b) as $\alpha(t+1)=\beta(t+1)=\kappa\alpha(t)$. Finally, the distributed iterative algorithm, namely Modified DDA-COV, is composed of Eqs. (18), (19) and (21) executed in steps $t=1, 2, \ldots$ until a stopping criterion is reached, e.g., $|r_s(t+1)\text{-}r_s(t)| < 0.001$. The algorithm is presented in Algorithm2.

In addition, since in DHD-CS the d_{sj} parameter is set before execution of Modified DDA-COV, there is no need for node s to send the value of r_s to its neighbors in each iteration, which results in decrease in the communication overhead compared to the DDA-COV algorithm.

References

Asada G, Dong M, Lin TS, Newberg F, Pottie G, Kaiser WJ. Wireless integrated network sensors: low power systems on a chip. In: Proceedings of the 1998 European solid state circuits conference. The Hague: Netherlands; 1998.

Bian F, Kempe D, Govindan R. Utility-based sensor selection. In: Proceedings of ACM IPSN; 2006.

Boukerche A, Fei X. A coverage-preserving scheme for wireless sensor network with irregular sensing range. Ad Hoc Networks 2007;5(8):1303–16.

Boyd S, Vandenberghe L. Convex optimization.New York, USA: Cambridge University Press; 2004.

Cardei M, Wu J, Lu M. Improving Network Lifetime using Sensors with Adjustable Sensing Ranges. International Journal of Sensor Networks 2006;1(1/2): 41–9.

Chu M, Haussecker H, Zhao F. Scalable information-driven sensor querying and routing for ad hoc heterogeneous sensor networks. International Journal of High Performance Computing Applications 2002;16(3):293–313.

CC2420 data sheet, Texas Instruments, March 2007.

Dhawan A, Vu CT, Zelikovsky A, Li Y, Prasad SK. Maximum lifetime of sensor networks with adjustable sensing range. In: Proceedings of the seventh ACIS international conference on software engineering, artificial intelligence, networking, and parallel/distributed computing (SNPD'06); 2006.

Edalat N, Xiao W, Tham C-K, Keikha E, Ong L-L. A price-based adaptive task allocation for wireless sensor network. In: Proceedings of IEEE MASS; 2009.

E2K-C data sheet, Long-distance Capacitive Proximity Sensor, Omron, Cat. No. D016-E2-04-X.

Esnaashari M, Meybodi M,R. A learning automata based scheduling solution to the dynamic point coverage problem in wireless sensor networks. Computer Networks 2010;54(14):2410–38.

Frank C, Romer K. Algorithms for generic role assignment in wireless sensor networks. In: Proceedings of ACM SenSys'05, San Diego, California: USA; 2005.

Ghosh A, Das S,K. Coverage and connectivity issues in wireless sensor networks: a survey. Pervasive and Mobile Computing 2008;4:303–34.

Gu Y, Ji Y, Li J, Zhao B. Qos-aware target coverage in wireless sensor networks. Wireless Communications and Mobile Computing 2009;9(12):1645–59.

Isler V, Bajcsy R. The sensor selection problem for bounded uncertainty sensing models. IEEE Transactions on Automation Science and Engineering 2006;3(4): 372–81.

Joshi S, Boyd S. Sensor selection via convex optimization. IEEE Transactions on Signal Processing 2009;57(2):451–62.

Kaplan L. Global node selection for localization in a distributed sensor network. IEEE Transactions on Aerospace and Electronic Systems 2006;42:113–35.

Karl H, Willig A. Protocols and architectures for wireless sensor networks.England: John Wiley & Sons; 2005.

Liu C, Cao G. Minimizing the cost of mine selection via sensor networks. In: Proceedings of the IEEE INFOCOM; 2009.

Miorandi D, Altman E, Alfano G. The impact of channel randomness on coverage and connectivity of ad hoc and sensor networks. IEEE Transactions on Wireless Communications 2008;7(3):1062–72.

Misra A, Eswaran S, La Porta T. Control-theoretic, mission-driven, optimization techniques for wireless sensor networks. in: Proceedings of the first international workshop on communication systems and networks (COMSNETS'09), Bangalore, India, pp. 1–8.

Naderan M, Dehghan M, Pedram H. Primal and dual-based algorithms for sensing range adjustment in WSNs. Journal of Supercomputing, in press-b, http://dx.doi.org/10.1007/s11227-011-0702-5.

Naderan M, Dehghan M, Pedram H, Hakami V. Survey of mobile object tracking protocols in wireless sensor networks: a network-centric perspective. International Journal of Ad Hoc and Ubiquitous Computing, in press-a.

Nakamura EF, Ramos HS, Villas LA, de Oliveira HABF, de Aquino ALL, Loureiro AAF. A reactive role assignment for data routing in event-based wireless sensor networks. Computer Networks 2009;53:1980–96.

Nanmaran A, Ponnavaikko M, Karthick N. Target tracking with integrated coverage in wireless sensor networks. In: Proceedings of the international conference on advances in computing, control and telecommunication technologies;2009: pp. 806–8.

Pathak A, Prasanna VK. Energy-efficient task mapping for data-driven sensor network macroprogramming. IEEE Transactions on Computers 2010;59(7): 955–68.

Photoelectric Sensors, 〈http://www3.panasonic.biz/ac/e/fasys/sensor/photoelectric/index.jsp〉.

Raghunathan V, Schurgers C, Park S, Srivastava MB. Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine 2002;19:40–50.

Rowaihy H, Sharanya E, Johnson M, Verma D, Bar-Noy A, Brown T, et al.. A survey of sensor selection schemes in wireless sensor networks. In: Proceedings of SPIE unattended ground, sea and air sensor technologies and applications IX, Orlando, FL, USA; 2007.

Rowaihy H, Johnson M, Sharanya E, Pizzocaro D, Bar-Noy A, La-Porta T, et al. Utility-based joint sensor selection and congestion control for task-oriented WSNs. In: Proceedings of Asilomar conference on signals, systems and computers; 2008.

Tian Y, Ekici E. Cross-Layer Collaborative In-Network Processing in Multihop Wireless Sensor Networks. IEEE Transactions on Mobile Computing 2007;6(3): 297–310.

Tselishchev Y, Boulis A, Libman L. Experiences and lessons from implementing a wireless sensor network MAC protocol in the castalia simulator. In: Proceedings of the IEEE WCNC, Sydney: Australia;2010: p. 1–6.

The Castalia Website. [Online]. Available from: 〈http://castalia.npc.nicta.com.au/〉.

The OMNet++ Website. [Online]. Available from: 〈http://www.omnetpp.org/〉.

Wang B. Coverage control in sensor networks.London: Springer-Verlag; 2010 Limited.

Wang H, Yao K, Pottie G, Estrin D. Entropy-based sensor selection heuristic for target localization. In: Proceedings of the third international symposium on information processing in sensor networks (IPSN), Berkeley: CA;2004: pp. 36–45.

Wang J, Medidi S.2007 Energy efficient coverage with variable sensing radii in wireless sensor networks. In: Proceedings of the IEEE third international

conference on wireless and mobile computing, networking and communications (WiMob); 2004:pp, 1–8.
Wang W, Srinivasan V, Wang B, Chua K-C. Coverage for target localization in wireless sensor networks. IEEE Transactions on Wireless Communications 2008;7(2):667–76.
Wu J, Yang S. Coverage Issue in sensor networks with adjustable ranges. In: Proceedings of ICPP; 2004:pp. 61–8.
Xie T, Qin X. An energy-delay tunable task allocation strategy for collaborative applications in networked embedded systems. IEEE Transactions on Computers 2008;57(3):329–43.