DDA-COV: A Distributed Dual-based Algorithm for Sensing Range Adjustment in WSNs

Marjan Naderan, Mehdi Dehghan and Hossein Pedram
Computer Engineering and Information Technology Department
Amirkabir University of Technology
Tehran, Iran
{naderan, dehghan, pedram}@aut.ac.ir

Abstract— In this paper, we have modeled the problem of adjusting the sensing range parameter for the application of multiple target coverage in wireless sensor networks (WSNs) through the Network Utility Maximization (NUM) framework. Our problem consists of maximizing the logarithm of sensing range of each sensor node as an approximation to the number of covered targets subject to energy constraint and reduction in the number of common covered targets. We assume the placement of sensors is known and the sensing range of each node is a continuous parameter. We used the dual decomposition technique to derive a distributed price-based algorithm, DDA-COV, for each node to adjust its sensing range during iterations in which the targets are static. Simulation results show the convergence of sensing ranges and utility function towards optimal values in few iterations with near optimal energy consumption of nodes.

Keywords- distributed algorithms; dual decomposition; utility function; sensing range; multi-target; coverage;

I. INTRODUCTION

Wireless sensor networks (WSNs) are networks in which nodes have the ability of sensing one (or more) physical phenomenon, and communicate wirelessly with limited capability of computation and battery resources. Monitoring and tracking of mobile targets have been marked as two major applications of these networks [1]. Tracking of mobile targets may include tracking of enemies, animals, humans and cars in highways. Some demo examples of real-life WSN surveillance and tracking systems include: a line in the sand [2], ExScal [3] and the ZebraNet [4] projects. Challenging protocols and services engaged in tracking problems in the layered architecture of communication networks are: scheduling [5] and MAC layer protocols [6], tracking protocols functioning in the network layer [7], coverage [8] and data aggregation [9].

In this paper we model a target coverage problem in WSNs in a slow, time-varying environment. Applications of such a scenario can be found in crowded places such as a downtown area of a city or a large natural habitat such as a national park [10]. In these scenarios, it is assumed that sensor nodes are located in areas with high densities of targets. By optimally adjusting the sensing ranges of the active nodes, the network covers the maximum number of targets and the energy consumption of sensor nodes is reduced. We model this problem in every time interval with static targets as an optimization problem maximizing total utilities of sensor nodes and derive a distributed algorithm to reach the globally optimal value via iterative methods.

The appropriate model for our problem is known as the Network Utility Maximization (NUM) framework which formulates a network system design problem as maximizing total utility of all the nodes subject to physical or economic constraints [11]. The NUM framework takes advantage of many advances in nonlinear optimization theory and distributed algorithms. In particular, the design of distributed algorithms in the NUM framework is based on primal and dual decomposition techniques [12]. Distributed solutions obtained within this framework attempt to reach the globally optimal value with iterative algorithms such as gradient and Newton’s method. Major characteristics of this framework that make it unique are [12]: the network is viewed as the optimizer, the optimization objective is the end-user application needs, the globally optimal performance benchmark is certified, and finally a widespread set of strategies for designing modularized and distributed solutions are obtained.

In this paper, we have used the dual decomposition method to solve our problem. According to [12] and [13], the basic idea of decomposition theory is to decompose the original large problem into sub-problems (at the lower layer) which are then coordinated by a master problem (at the higher layer) by means of some kind of signaling, often without the need to solve the master problem centrally. Our optimization problem is the maximization of the sum of utilities of nodes, defined as the number of covered targets by each node, subject to energy and overlapping sensing range constraints.

Previous works on the target coverage problem or point coverage aim at finding the best sensor placement to provide uniform coverage of targets [14,19], while we seek to optimize the sensing range parameter. In addition to the works presented in [14] and [19], Cardei et. al. have used sensors with a finite set of sensing ranges with their corresponding energy consumption levels to cover all targets in a sequence of set covers in [15] and [16].

A recent study in [17] has investigated the joint problem of coverage and identity management for multi-target tracking in WSNs. The authors have introduced the eTrack protocol to...
tackle this problem. In [18], the problem of analyzing the one-dimensional path coverage by an area coverage process is introduced, which estimates the position of a target as a function of time. In [19], Baumgartner and Ferrari studied the geometric properties of the network, addressing a deterministic track-coverage formulation using binary sensor models. In [20], Lin and Tang proposed a grid-based approach to provide maximum coverage for a mobile target in a hybrid sensor network with mobile sensor nodes in a distributed fashion.

Our work differs from Cardei’s works in [15] and [16], in the sense that we have assumed the sensing range parameter as a continuous variable to be optimized in our NUM-based Non-Linear Programming (NLP) problem. In the above works the problem is to find sensor placements while in our study we aim to find optimal sensing ranges of nodes. Furthermore, we use the dual-based decomposition technique to solve our optimization problem which has not been used in any of the previous coverage-related studies.

Usage of the NUM framework has attracted the research community during the last decade and since the introduction of primal and dual algorithms by Kelly’s seminal paper in 1998 [21]. In [21], the problem of optimizing the source rates, SYSTEM, is divided into two sub-problems USER and NETWORK. In [22], Low and Lapsley solved the rate control problem via its dual problem. A survey of decomposition algorithms can be found in [12].

In wireless sensor networks several works have been recently incorporating various parameters with or without the traditional “rate” parameter in the basic NUM problem. In [23], the energy parameter is considered in the NUM problem; coupled in the objective function and in the constraints. In [24], the rate allocation problem is combined with network lifetime. In [25], the authors have proposed the NUM model with constrained delay. In [26], Baras et. al. have extended the NUM problem formulation to encompass security measures such as trust. In [11], joint maximization of sampling rate with dynamic routing is investigated.

In addition, some extensions to the original work of Kelly have been proposed. In [27], the authors have extended the model for multicast receivers in wireless networks by adapting the capacity constraint of links to account for contention. In [28], the authors have combined the original rate control problem with a selection strategy by deriving a Mixed Integer Programming (MIP) problem. None of the above researches has considered the target coverage problem modeled in the NUM framework and has solved it via dual decomposition.

The rest of this paper is organized as follows: in section II, we define the assumptions and formulation of our problem. In section III, we solve the optimization problem by solving the dual problem. In section IV, we present our simulation results. Finally, in section V, we conclude the paper.

II. ASSUMPTIONS AND PROBLEM FORMULATION

We want to find the optimal sensing range \( r_i \) of sensor nodes in a network with a dense placement of targets. It is assumed that the targets move moderately between neighboring sensors and there exist always some targets around each sensor node. We divide the total duration of the target tracking network into time intervals \( T_i \), in which, targets are assumed to be static in these intervals and each interval consists of a number of iterations \( t_i \), such that the network converges to its optimal values.

We define the utility of a node, \( U_c(\cdot) \), for the network to be an approximation of the number of targets it covers. Since the number of targets which a sensor node covers grows with increasing \( r_i \) (until \( r_i = r^{\text{max}}_i \)), any increasing, continuous differentiable function of \( r_i \) can be used for \( U_c(\cdot) \). We assume \( U_c(r_i) = \log(r_i) \), a concave function, which is additionally consistent with the concept of “locality of movement” for targets in [10]. Obviously, we must have \( r^{\text{min}}_i \leq r_i \leq r^{\text{max}}_i \) and \( 1 < r^{\text{min}}_i \), such that \( U_c(r_i) > 0 \).

With the above utility function, the sensing radius is assumed a continuous parameter and a sensor \( s \) is able to dynamically adjust its sensing range \( r_j \) by any arbitrary amount (as long as \( r_i > r^{\text{max}}_i \)), i.e. each sensor’s sensing range is elastic. This assumption does not influence the basic behavior of the NUM framework despite it may not be completely pertinent to sensors with a finite set of sensing ranges [27].

On the other hand, increasing the sensing range of a node \( s \) increases the energy consumption of that node. In general, the energy consumption model of the sensing range does not necessarily obey that of the communication range [29]. In this paper, we assume energy consumption related to the sensing operation to be a linear function, consistent with the assumptions in [15] and [16]. Thus, the consumed energy of sensing is \( e^\text{sensing} = A r_s \), and it must be less than its remaining energy for an active node: \( e^\text{remaining}(s) < e^\text{sensing} \).

Another consequence of increasing the sensing range is the growth in the number of common targets between neighbor nodes. In fact, overlapping sensing ranges of neighbor nodes are more likely to cover common targets. Thus, we require sensor node \( s \) to have sensing range \( r_s \), related to that of its neighbors by \( r_s + r_j \leq d_{ij} \) for all \( j \in \text{NBR}(s) \), where \( d_{ij} \) is the Euclidean distance between neighbor nodes \( s \) and \( j \). Despite that this constraint leaves some parts of the area to be uncovered, it is consistent with our logarithmic assumption of \( U_c(r_j) \), which implies more targets are near and a few of them are far from sensor nodes.

Our target tracking network consists of a set of sensor nodes \( S | | S | = N \), and a set of moving targets, \( T | | T | = M \). Each sensor node \( s \), with sensing range \( r_s > r^{\text{min}}_s \), covers a subset of targets, \( \text{Target}(s) \neq \emptyset \). We have used the Boolean sensing model [14], where targets within sensing ranges \( r_s \) (\( r^{\text{min}}_s \leq r_s \leq r^{\text{max}}_s \)) are detected reliably (with probability one) and targets outside this range are not detected at all. The set of neighbor nodes of \( s \) is denoted as \( \text{NBR}(s) \) and each sensor node \( s \) has a finite supply of remaining energy, \( e^\text{remaining}(T) \) at the beginning of time interval \( T_i \).

According to the above, the NUM problem is considered as:

\[
\text{NUM-COV}(U, A, E^\text{remaining}(T)):
\]

\[
\begin{align*}
\text{Maximize} & \quad \sum_{ij} U_c(r_i) \\
\text{st.:} & \quad e^\text{sensing}_s = A r_s \leq e^\text{remaining}(T) \quad s = 1, \ldots, N \\
& \quad r_s + r_j \leq d_{ij} \quad s = 1, \ldots, N, \forall j \in \text{NBR}(s) \\
& \quad r^{\text{min}}_s \leq r_s \leq r^{\text{max}}_s
\end{align*}
\]

where \( U, A \) and \( E^\text{remaining}(T) \) are vectors of \( U_c, A \), and \( e^\text{remaining}(T) \) for \( s = 1, \ldots, N \), respectively. We have denoted (1) with \( \text{NUM-COV}(U, A, E^\text{remaining}(T)) \) which shows \( E^\text{remaining} \) changes with time and \( \text{NUM-COV}(U, A, E^\text{remaining}) \) in the rest of this paper as it is solved in each times interval.
In addition, for each sensor node \( s \) with location \( l_s \) and for all the targets \( k \) with locations \( T_k \) in its sensing range \( r_s \), \( k \in \text{Targ}(s) \), we must check the condition \( \|l_s - T_k\| \leq r_s \). Otherwise, it is not a valid configuration. Problem (1) is convex NLP since, the objective function is concave and the constraints are linear.

It is obvious that (1) can be solved centrally only if the network (planner) knows the utility functions of all sensors, or if there is a motivation for the sensors to disclose their utility functions truthfully [30]. In this paper we take the dual decomposition approach to achieve a distributed algorithm which reaches the globally optimal values iteratively.

III. DUAL-BASED SOLUTION FOR NUM-COV(\( U,A,E_{zem} \))

According to [12] and [13], a dual decomposition is appropriate when the problem has a coupling constraint such that, when relaxed, the optimization problem decouples into several sub-problems. Hence, we continue with the Lagrangian of (1) with a general utility function:

\[
L_{\text{NUM-COV}}(r, \mu, \gamma) = \sum_{s \in S} U_s(r_s) + \sum_{s \in S} \mu_s (e^s_{\text{con}} - A r_s)
\]

\[
+ \sum_{j: j \in \text{NBR}(s)} \gamma_j (d_{ij} - r_j)
\]

where \( \mu \) is the vector and \( \gamma \) is the matrix of Lagrange multipliers with their interpretations as energy prices and overlapping prices, respectively. We can rewrite it as:

\[
L_{\text{NUM-COV}}(r, \mu, \gamma) = \sum_{s \in S} \left( U_s(r_s) - \mu_s A r_s - r_s \sum_{j: j \in \text{NBR}(s)} \gamma_j \right)
\]

\[
+ \sum_{s \in S} \left( e^s_{\text{con}} \mu_s + \sum_{j: j \in \text{NBR}(s)} \gamma_j (d_{ij} - r_j) \right)
\]

From the above expression, the Lagrangian can be separated into many sub-problems (at the lower level), i.e.,

\[
\max \sum_{s \in S} \left( U_s(r_s) - \mu_s A r_s - r_s \sum_{j: j \in \text{NBR}(s)} \gamma_j \right) = \sum_{s \in S} \max \left( U_s(r_s) - \mu_s A r_s - r_s \sum_{j: j \in \text{NBR}(s)} \gamma_j \right)
\]

The dual function is (for each sensor \( s \)):

\[ D(\mu_s, \gamma_j) = \text{Maximize} \ L_{\text{NUM-COV}}(r_s, \mu_s, \gamma_j) \]

\[ \text{s.t.:} \ r_s^{\text{min}} \leq r_s \leq r_s^{\text{max}} \]

and the master dual problem (at the higher level) is:

\[ \text{Minimize} \ D(\mu, \gamma) \]

\[ \text{s.t.:} \ \mu_s \geq 0, \ \gamma_j \geq 0 \]

Since \( U_s \) is concave and the constraints in (1) are linear there is no duality gap and dual optimal prices, which are Lagrange multipliers, exist [22]. Hence, we first solve (6). To obtain dual optimal prices, we use a relaxation to design dynamic algorithms that asymptotically (in time) approach the required maximum [30]. A natural candidate for such an algorithm is the gradient descent algorithm from optimization theory [31,32]. According to this algorithm, to update the Lagrange multipliers we have:

\[
\mu(t+1) = \left[ \mu(t) - \alpha(t) \left( e^s_{\text{con}} - A r_s(t) \right) \right] \]

\[
\gamma_j(t+1) = \left[ \gamma_j(t) - \beta(t) (d_{ij} - r_j(t)) \right]
\]

where \([\cdot]^t\) denotes the projection onto the set \( R^+ \) of non-negative real numbers, \( t \) is an iteration in the interval \( T \), as \( t \in T = [t_1, t_2, ..., t_n] \), and \( \alpha(t) \) and \( \beta(t) \) are positive scalar step sizes.

Now that we have obtained \( \mu^* \) and \( \gamma^* \), the primal optimal \( r^* \) can be computed by individual nodes \( s \) locally. The primal optimal \( r \) is obtained by considering (5). Since the Lagrangian is separable, this maximization of Lagrangian over \( r_s \) can be conducted in parallel at each source \( s \) by:

\[
\text{Maximize} \ U_s(r_s) - \mu_s A r_s - r_s \sum_{j: j \in \text{NBR}(s)} \gamma_j
\]

\[ \text{s.t.:} \ r_s^{\text{min}} \leq r_s \leq r_s^{\text{max}} \]

To solve (9), each node takes the derivative of the objective function of (9) by assuming \( U_s(r_s) = a_s \log(r_s) \), which results in:

\[
\frac{\mu_s A_s + \sum_{j: j \in \text{NBR}(s)} \gamma_j}{r_s^{\text{min}}} \quad \text{if} \ r_s^{\text{min}} \leq r_s \leq r_s^{\text{max}}
\]

\[
\frac{\mu_s A_s + \sum_{j: j \in \text{NBR}(s)} \gamma_j}{r_s^{\text{max}}} \quad \text{if} \ r_s^{\text{max}} > r_s \geq r_s^{\text{min}}
\]

An important consideration to this approach in comparison with Kelly’s algorithm [21] is that in [21] the non-negativity constraint on the source rates and link prices are relaxed in the primal solution, which allows stability proof via a Lyapunov function. In our approach, the dual objective function, \( D(\mu_s, \gamma_j) \), can be viewed as a Lyapunov function (due to the projection to the positive quadrant), provided the step sizes are sufficiently small [22]. Hence, the following theorem results:

**Theorem:** by equations (7) and (8), dual variables \( \mu_s(t) \) and \( \gamma_j(t) \) converge to the optimal dual solutions \( \mu^* \) and \( \gamma^* \) if the step sizes are chosen such that

\[
\alpha(t) \to 0, \sum_{i=1}^{\infty} \alpha(i) = \infty \quad \beta(t) \to 0, \sum_{i=1}^{\infty} \beta(i) = \infty
\]

At this stage we can present the distributed dual-based coverage algorithm DDA-COV as below. The algorithm is stopped when a stopping criterion is reached [32], e.g., when the value remains the same for a number of iterations. In our simulation experiments \( r_s(t+1) = r_s(t) \) when \( |r_s(t+1) - r_s(t)| < 0.001 \) and we run this algorithm for a predefined number of iterations.

**Algorithm DDA-COV:** Distributed Dual-based

**Algorithm for NUM-COV(\( U,A,E_{zem} \))**

**Initialization:** set \( r_s(0) = r_s^{\text{min}} \) for \( s = 1, ..., N \), and the values of \( \mu_s(0), \gamma_j(0), \alpha(0) \) and \( \beta(0) \) to arbitrary positive values.

At each iteration \( t = 1, 2, ..., \):

1) Each node \( s \) locally updates its energy and overlapping prices according to (7) and (8), respectively. Note that each node \( s \) has a distance price \( \gamma_j(s) \) for every neighbor node \( j \) at each iteration \( t \).

\[
\text{Algorithm DDA-COV: Distributed Dual-based}
\]

701
Each node $s$ can update its prices by its local information.

2) Each node $s$ locally finds its new sensing range, $r_s(t+1)$, for the next iteration by (10), and for $\mu_s = \mu_s(t)$ and $\gamma_s = \gamma_s(t)$.

3) Each node $s$ communicates its new sensing range $r_s(t+1)$ with its neighbors.

IV. SIMULATION RESULTS

In this section we explain a series of simulation experiments to test the convergence of the DDA-COV algorithm and its energy consumption. We compare our results with optimal values computed by solving the centralized NLP formulation of (1). We used MATLAB 7.8.0 [33] and AIMMS 3.9 [34] to implement our algorithm and find optimal values, respectively. Moreover, in spite that our algorithm has no limitations for the network structure and the number of nodes, we have used grid network structure only to simplify the exchange of values between neighbor nodes.

A. Determining the initial and constant values of our algorithm

There are some constant values in our algorithm that need to be determined correctly. Values of step sizes, $\alpha(t)$ and $\beta(t)$, are determined according to the inexact line search algorithm in [32] and with respect to (11) (the Theorem), i.e., $\alpha(t+1)=\beta(t+1)=\kappa \alpha(t)$.

To choose a value for $\kappa$ we tested the convergence of $\mu$ in (9) with different values of $\kappa$ in range $[0.1, 0.9]$ for a sample node versus iterations as in Fig. 1. Other nodes have nearly the same amount and trend in changes of $\mu(t)$.

As seen in Fig. 1.a, $0.1 \leq \kappa < 0.5$ results in large values of $\mu$ and consequently wrong values for $r_s$ in (10). For $\kappa=0.5$ and $0.5 \leq \kappa \leq 0.9$ (Fig. 1.b), $\mu(t)$ converges in less than 10 iterations with nearly no oscillations. Values in Fig. 1.b are very close to each other thus, we have shown only the part of the graph with differences. According to this experiment, we have obtained results for both cases of $\kappa=0.5$ and $\kappa=0.7$ in the rest of our experiments.

We have tested the algorithm with two cases for initial energy of nodes: nodes having the same initial energy, and nodes having different initial energies. There is only a slight difference in the rate of convergence between these two cases and hence, we assume nodes have different initial energies. Table I shows a sample of initial energies for a 40x40 m² grid network with 16 nodes. Targets are placed near the sensor nodes such that at optimal ranges complete coverage is guaranteed. Sensing ranges of all nodes vary between 1.1 and 6 meters and the communication range is set to 10 meters. We run the algorithm for 40 iterations, despite 30 iterations is also sufficient for convergence of $\mu(t)$ and $\gamma(t)$.

B. Convergence of the sensing ranges

With the values defined in the previous subsection, we tested the DDA-COV algorithm for a grid network with 16 nodes. Fig. 2 shows the values of the sensing ranges vs. iterations for $\kappa=0.5$ and $\kappa=0.7$ only for nodes 9-12 (due to space limitations). As seen from Fig. 2, there are more oscillations during initial iterations for $\kappa=0.7$ but finally the algorithm converges near to the optimal values ($r_s=5m$).

Fig. 3.a and Fig. 3.b show total utility function and remaining energies of nodes, respectively versus iterations for the two cases $\kappa=0.5$ and $\kappa=0.7$. As seen from Fig. 3.a, there are more oscillations during initial iterations for $\kappa=0.7$ but finally the algorithm converges to the same results. Optimal value of the utility function, shown in Fig. 3.a, is 257.5 and it is reached in less than 10 iterations by our DDA-COV

### Table I. Initial energies of nodes for a grid network

<table>
<thead>
<tr>
<th>Node’s IDs</th>
<th>Nodes coordinate</th>
<th>Initial energy (Joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4</td>
<td>(10, 10), (20, 10), (30, 10), (40, 10)</td>
<td>30</td>
</tr>
<tr>
<td>5, 6, 7, 8</td>
<td>(10, 20), (20, 20), (30, 20), (40, 20)</td>
<td>25</td>
</tr>
<tr>
<td>9, 10, 11, 12</td>
<td>(10, 30), (20, 30), (30, 30), (40, 30)</td>
<td>30</td>
</tr>
<tr>
<td>13, 14, 15, 16</td>
<td>(10, 40), (20, 40), (30, 40), (40, 40)</td>
<td>40</td>
</tr>
</tbody>
</table>
algorithm for $\kappa=0.5$. From Fig. 3.b, it can be deduced that $\kappa=0.5$ is more energy conserving than $\kappa=0.7$ and it is closer to the optimal value. Optimal value of remaining energies of nodes in Fig. 3.b is calculated centrally (at one iteration); hence, it is higher than the values reached by our algorithm.

C. Increasing the number of nodes

In this series of simulations, we evaluated our distributed algorithm by increasing the number of nodes. The number of targets and the area also increase as mentioned in Table II. Fig. 4 depicts total utility function of the network versus number of nodes in networks with grid structures. As seen from Fig. 4.a, total utility function increases as the number of nodes increase. Values of the total utility function for $\kappa=0.5$ and $\kappa=0.7$ are very close to each other since, the only difference is due to some oscillations for $\kappa=0.7$. They are also close to the optimal value in Fig. 4.a, in which more details can be viewed in Fig. 4.b at $N=25$.

Fig. 5 depicts total remaining energies of nodes versus number of nodes. It can be seen from Fig. 5.a, that total remaining energies of nodes increases with increase in the number of nodes. The difference can be viewed in more detail in Fig. 5.b where the values of energy consumption for DDA-COV with $\kappa=0.5$ are very close to the optimal ones.

![Figure 3](image.png)

**Figure 3.** Changes of a) total utility function and b) total remaining energies, of nodes in a network of 16 nodes during iterations.

![Figure 4](image.png)

**Figure 4.** Changes of total utility function vs. number of nodes, a) values are very close to each other, b) in more detail.

![Figure 5](image.png)

**Figure 5.** Changes of total remaining energies vs. number of nodes, a) values are very close to each other, b) in more detail.

![Figure 6](image.png)

**Figure 6.** Changes of number of iterations needed for convergence versus number of nodes in networks with grid structures. An important corollary resulting from this figure is that for a specific value of $\kappa$, the number of iterations needed for convergence does not change drastically as the number of nodes increase.

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>Area $(m^2)$</th>
<th>Number of targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>20x20</td>
<td>12</td>
</tr>
<tr>
<td>9</td>
<td>30x30</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>40x40</td>
<td>40</td>
</tr>
<tr>
<td>25</td>
<td>50x50</td>
<td>60</td>
</tr>
<tr>
<td>36</td>
<td>60x60</td>
<td>84</td>
</tr>
<tr>
<td>49</td>
<td>70x70</td>
<td>112</td>
</tr>
</tbody>
</table>

**Table II.** Number of targets in networks with different number of nodes

**Fig. 6.** Shows the number of iterations needed for convergence versus number of nodes in networks with grid structures. An important corollary resulting from this figure is that for a specific value of $\kappa$, the number of iterations needed for convergence does not change drastically as the number of nodes increase.
nodes is increased. For $κ=0.7$, more iterations are needed compared to $κ=0.5$, which is obviously due to more oscillations occurring in this case. This fact shows that our DDA-COV algorithm scales well in terms of number of iterations needed for convergence as the number of nodes in the network increases.

D. A scenario with moving targets

In these simulations we changed the locations of some targets in three consecutive time intervals. In the last interval some targets are out of range of any sensor and we have partial coverage. We used the DDA-COV with $κ=0.5$ in each time interval with 30 iterations.

Fig. 7 shows the convergence of sensing ranges for nodes 9-12 during these three consecutive time intervals. Fig. 8.a and Fig. 8.b show total utility function and the reduction in the remaining energies of nodes during three time intervals. It can be seen that the total utility function reaches its maximum value in each time interval from Fig. 8.a. The reduction of remaining energies also can be seen in Fig. 8.b which reflects $E_{rem}(T)$ as presented in (1).

V. CONCLUSION AND FUTURE WORK

In this paper we investigated the applicability of the NUM framework for a target coverage problem for multi-target tracking in WSNs. We model the problem of finding the optimal sensing ranges in each time interval with static targets as a convex non-linear formulation and solve it via dual decomposition with the gradient ascent algorithm to reach the optimal values. Simulations results show the convergence of our dual-based distributed algorithm, DDA-COV, to optimal values, in addition to the near optimal energy consumption of our algorithm, especially for the case $κ=0.5$.

We are currently working on the primal solution to (1) and its stability proof via a Lyapunov function. Another possibility is to look into other models for $e^n_{t,x}$. Including constraints for comparison of remaining energies of neighbor nodes is another extension which requires (1) to be rewritten and the feasibility of any other primal or dual approaches to be investigated. Furthermore, we aim to investigate the problem of joint selection and coverage with the NUM framework in which, not all the sensors are active.

ACKNOWLEDGMENT

The authors would like to thank Ms. Shirin Ghanbari (Ph.D.) and Mr. Mahmoud Naderan (Ph.D. student) for their helpful comments on the paper.

REFERENCES


[34] http://www.aimms.com/