Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 4602 – 4614

doi:10.3906/elk-1702-107

Lifetime maximization of wireless sensor networks with sink costs

Muhammed Emre KESKİN*
Department of Industrial Engineering, Faculty of Engineering, Atatürk University, Erzurum, Turkey

Received: 09.02.2017 • **Accepted/Published Online:** 13.07.2017 • **Final Version:** 03.12.2017

Abstract: Balanced distribution of the energy load of the sensors is important for the elongation of wireless sensor network (WSN) lifetimes. There are four main WSN design issues affecting the energy distribution among the sensors: sensor locations, sensor activity schedules, mobile sink routes, and data flow routes. Many studies try to make energy usage more efficient through optimal determination of these design issues. However, only very few studies handle these four design issues in a combined manner. Additionally, the cost of the sinks is neglected in all studies. In this study, a mixed integer linear program is first proposed, in which the cost of the sinks is taken into consideration in terms of lifetime hours and the four design issues are integrated. Next, a heuristic solution procedure for the solution of large network instances is offered and the efficiency of the heuristic is proven by comparing its performance with that of commercial solvers in extensive numerical instances.

Key words: Wireless sensor networks, mixed integer linear program, sink costs

1. Introduction

Wireless sensor networks (WSNs) consist of many small, electronic, multifunctional devices that are deployed over a region of interest, called the sensor field. A sensor senses the environment within its sensing range and the collected data are transferred to gateway nodes called the sink, either directly or through other sensors. Transmission is possible only if the receiver sensor or the sink lies in the communication range of the transmitting sensor. Sensors collectively function to produce a distributed environment, which enables monitoring of remote or hostile environments. This is the main reason for the wide applicability of WSNs [1]. Sensors can be in active or standby modes. A sensor in the active mode senses its environment, receives data from neighboring sensors, and transmits the data to its neighboring sensors or directly to the sink if the sink is nearby. On the other hand, a sensor in the standby mode consumes negligible energy.

There are four main WSN design issues: 1) sensor locations, 2) sensor activity schedules, 3) sink locations or routes, and 4) data flow routes. Sensor locations should be determined so that the coverage demands of the sensor field are satisfied. This problem is named the coverage problem (CP). It should be noted that the sensor deployment issue can be a decision subject only if the terrain that the WSN will be deployed on is under full control. In many applications, precise location determination would be very costly or simply impossible. For instance, sensor locations should be arbitrarily determined for a WSN deployed on the ocean floor for monitoring seismic activity. Similarly, one may not have the luxury of precise sensor placement for military WSN applications deployed in hostile environments. On the other hand, it is possible to determine candidate sensor locations for a WSN application designed for a large greenhouse for monitoring the conditions of the

*Correspondence: emre.keskin@atauni.edu.tr

planted vegetables. In WSN design studies, coverage demands of the sensor field are generally characterized by forming a set of discrete coverage points and appointing coverage demands to these points. Since some regions of the field can be more or less important, demands of the coverage points can be different, implying a heterogeneous CP. Determination of the activity schedules is known as the activity scheduling problem (ASP). Sensors having low residual energies can be put into standby mode; at the same time, some of the sensors that are in standby mode can be made active. However, the number and distribution of the active sensors in any period of the lifetime should be able to meet the coverage requirements. The third WSN design issue is the sink placement problem (SPP). Sink locations significantly affect sensor energy loads. Sensors near the sinks, called relay sensors, spend more energy than the rest of the sensors, since all the data produced by the network flow through relay sensors. Therefore, relay sensor batteries deplete faster, which leads to the isolation of the sinks from the rest of the network. This phenomenon is called “the crowded center effect” in [2], “the energy hole problem” in [3,4], and “the sink neighborhood problem” in [5]. Changing the set of relay sensors by controlled sink mobility is offered as a remedy for this problem. SPP is called the sink routing problem (SRP) when the sinks are mobile. The fourth design issue is related to data routes. Energy is mostly spent during data transmissions in WSNs. Hence, data routes should be carefully selected. This problem is known as the data routing problem (DRP). Another design issue that can be considered along with the above mentioned issues is sink cost. As far as we know, only the number of the sinks is considered as a parameter in all the studies in the literature. There is no study that concentrates on finding the number of sinks that leads to maximum lifetime and, at the same time, costs as little as possible. In this study, sink costs are taken into consideration by making use of a parameter representing the cost of a sink in terms of lifetime hour.

It should be emphasized that WSN design studies affect each other, implying that they should be handled together. However, most of the studies from the literature concentrate on only a single design issue or a subset of four design issues. For instance, only the CP is studied in [6]. Similarly, when [5,7–9] try to solve the SRP, it is assumed that sensor locations and activity schedules of the sensors are given, and that data flow through shortest paths. There are many studies that combine SRP and DRP. One of the earliest of these studies is [10], in which two models are presented wherein total used energy and the maximum energy are minimized. The same problem is handled in [11], but the difference between the maximum and minimum energy usages is minimized. The framework of [12] is also similar, but it maximizes minimum residual energy. The same framework is borrowed in [13] but the perspective is extended so that multiple time periods are integrated. In contrast, [14] directly maximizes the lifetime under energy usage and flow balance constraints; [15] analyzes the model of [14] but offers a distributed solution strategy depending on Lagrangian relaxation. Similarly, [16] extends the same model for delay-tolerant applications. In addition, [17] and [18] come up with distributed solution strategies for the delay-tolerant model of [16]. In addition to these, [19] extends the model of [14] by integrating the multiple static sink placement problem. Finally, [20] develops a model including multiple mobile sinks. Studies combining more than two design issues are rare. One of them is [21], which develops [14] by also integrating the CP. ASP is combined on top of the model of [21] in studies [22–24]. A disjoint set heuristic is offered in [22], while in [23] a heuristic based on Lagrangian relaxation is developed. The solution method of [24] depends on column generation. Alternatively, the models in [25] are improved in [22–24] by making the sinks mobile. In [26], design issue integration’s effect on network lifetime is analyzed via an empirical approach. Finally, [27] a column generation algorithm is provided for a mathematical model in which the WSN design issues are integrated, while [28] offers a convex optimization problem for determining optimal topology for network-based WSNs. In this study, CP, ASP, SRP, and DRP are integrated in a mixed integer linear program (MILP). The sink costs are represented in terms of network lifetime hours in the objective function. In addition, a heuristic solution strategy to solve large instances is offered.

The rest of the paper is organized as follows. The MILP model integrating CP, ASP, SRP, and DRP continues in Section 2. Section 3 introduces the heuristic solution strategy and the numerical results of the heuristic and the commercial solver Gurobi are given in Section 4. Finally, the paper is summarized and future research directions are proposed in Section 5.

2. MILP

For better visualization of the problem, a sample instance with 30 candidate sensor locations and 15 candidate sink visit points is given in Figure 1. It is assumed that there are two periods, and the locations of the sensors, sinks, and data routes are represented by arrows in both periods. There are sensors that are in active mode in both periods, while some of them are in standby mode in one period and active in the other. Sensing and communication ranges of the sensors are taken as one for the instance, implying that each of the coverage requirements (which are taken as two) are met, since there are at least two sensors capable of sensing each point in both periods. Another assumption that can be made is that some sensors are able to send their data to a sink directly, while some others send their data through other sensors.

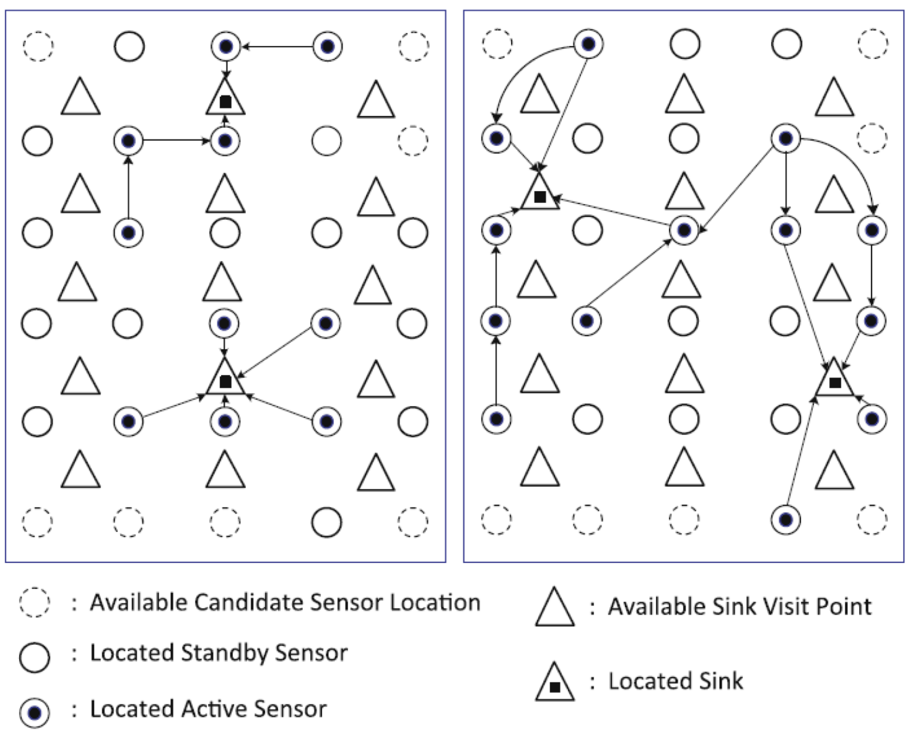

Figure 1. A sensor network with sensors, mobile sinks, and data flows.

The coverage requirements of the sensor field are represented by a set of points called K. Candidate sensor and sink locations are represented by sets S and L, respectively. A set of sinks is named N, the set of sensor types is given by R, and the set of periods is named T. A sensor of type $r \in R$ located at point $i \in S$ is referred as sensor (i,r) in the sequel. The set of candidate sink and sensor locations in the transmission range of sensor (i,r) are denoted by S_{ir} and N_{ir}, respectively, while the set of coverage points in the sensing range of sensor $(i, r)$is represented by K_{ir}. Parameters are defined depending mostly on the above described sets. For instance, d_k stands for the requirement of point $k \in K$, while e_r and h_r, respectively, represent the initial battery energy and unit data generation rate of a sensor of type r. Similarly, c^r and c^s represent the energy segments used for receiving a unit bit of data, and sensing and processing a unit bit of data per unit time, respectively. Additionally, c^t_{ij} represents the energy spent transmitting a unit bit of data between a sensor located at point i and a sensor or sink located at point j. b_{ir} is the cost of locating sensor (i, r) and B is the total budget allocated for sensor deployment. There are also big M parameters denoting large numbers. Finally, β represents the cost of a sink in terms of lifetime hours. There are eight sets of decision variables in the formulation. w_t stands for the length of period t. x_{irjst} represents the data transferred from sensor (i, r) to sensor (j, s) in period t while y_{irlt} is the amount of data transmitted from sensor (i, r) to the sink located at sink visit point l in period t. Note that data routing decisions are incorporated into the formulation with the x_{irjst} and y_{irlt} decision variables. Additionally, a_{irt} is equal to the amount of time the sensor (i, r) is active in period t, p_{ir} indicates whether or not sensor $(i, r)$is located, and q_{irt} shows whether or not sensor $(i, r)$is active in period t. Sensor deployment decisions and sensor activity schedules are incorporated into the model by decision variables p_{ir} and q_{irt}, respectively. z_{lt} is equal to 1 if there is a sink at visit point l in period t, and zero otherwise. Sink movement decisions are incorporated into the model with the decision variable z_{lt}. Finally, θ_n indicates whether or not sink n is used. The MILP model integrating CP, ASP, SRP, and DRP (called CASD) is given as:

$$\max \sum_{t \in T} w_t - \beta \sum_{n \in N} \theta_n \tag{1}$$

subject to:

$$\sum_{s \in R} \sum_{j: i \in S_{js}} x_{jsirt} + h_r a_{irt} = \sum_{l \in L_{ir}} y_{irlt} + \sum_{s \in R} \sum_{j: i \in S_{js}} x_{irjst} \qquad i \in S r \in R t \in T \tag{2}$$

$$\sum_{t \in T} \left(c^s a_{irt} + c^r \sum_{s \in R} \sum_{j: i \in S_{js}} x_{jsirt} + \sum_{l \in L_{ir}} c^t_{il} y_{irlt} + \sum_{s \in R} \sum_{j: i \in S_{js}} c^t_{ij} x_{irjst} \right) \le e_r \qquad i \in S r \in R \tag{3}$$

$$\sum_{r \in R} \sum_{i: l \in L_{ir}} y_{irlt} \le M z_{lt} \qquad l \in L t \in T \tag{4}$$

$$\sum_{l \in L} z_{lt} = \sum_{n \in N} \theta_n \qquad t \in T \tag{5}$$

$$\sum_{r \in R} \sum_{i: k \in K_{ir}} q_{irt} \ge d_k \qquad k \in K t \in T \tag{6}$$

$$\sum_{i \in S} \sum_{r \in R} b_{ir} p_{ir} \le B \tag{7}$$

$$q_{irt} \le p_{ir} \qquad i \in S r \in R t \in T \tag{8}$$

$$\sum_{s \in R} \sum_{j \in S_{ir}} x_{irjst} \le M q_{irt} \qquad i \in S r \in R t \in T \tag{9}$$

$$\sum_{s \in R} \sum_{j: i \in S_{js}} x_{jsirt} \leq M q_{irt} \qquad i \in Sr \in Rt \in T \tag{10}$$

$$a_{irt} \leq w_t \qquad i \in Sr \in Rt \in T \tag{11}$$

$$a_{irt} \leq M q_{irt} \qquad i \in Sr \in Rt \in T \tag{12}$$

$$a_{irt} \geq w_t - M(1 - q_{irt}) \qquad i \in Sr \in Rt \in T \tag{13}$$

$$w_t x_{irjst}, y_{irlt}, a_{irt} \geq 0 \tag{14}$$

$$p_{ir} q_{irt}, z_{lt}, \theta_n \in \{0, 1\} \tag{15}$$

Lifetime is defined as the summation of the period lengths and it is maximized in the objective function. It is assumed that each of the sinks used has a cost corresponding to β lifetime hours, and so the number of used sinks is multiplied by β and subtracted from the lifetime. In constraint (1), the total incoming data to sensor (i,r) is summed up with the data generated by sensor (i,r) and equated to the total outgoing data from sensor (i,r) for each period. In constraint (2), the energy used by sensor (i,r) for sensing and processing data, for receiving energy from neighboring sensors, and for transmitting data to neighboring sensors and sinks are summed up over periods and forced to be less than or equal to the initial battery energy. Constraint (3) prevents a sensor from sending data to a sink visit point if there is no sink located there. Constraint (4) ensures that the number of located sinks is equal to the number of selected sinks in each period. Constraint (5) makes sure that the number of active sensors that are close enough to sense each coverage point is equal to its coverage demand. Constraint (6) limits the total sensor deployment cost by on-hand budget. Constraint (7) switches undeployed sensor activity to standby throughout the lifetime. Constraints (8) and (9) equate the total outgoing data and total incoming data, respectively, to zero if the sensor is not active. Constraints (11–13) are technical constraints defining the a_{irt} variable. Namely, a_{irt} variable is equal to the amount of time that sensor (i,r) is active in period t, i.e. $a_{irt} = q_{irt} w_t$. If sensor (i,r) is active in period t, namely $q_{irt} = 1$, a_{irt} is equated to w_t with constraints (10) and (12). Similarly, if sensor (i,r) is in standby mode in period t, namely $q_{irt} = 0$, a_{irt} is equal to zero with the help of constraint (11) and the nonnegativity restriction (13). Finally, constraints (13) and (14) are nonnegativity restrictions on continuous variables of the model and binary restrictions on the rest of the variables, respectively.

3. Solution method

General MILP problems belong to the NP-hard problem class, and the present problem formulated as CASD is no exception. If sensors are kept in active mode throughout their lifetime, if sinks are static and placed at predetermined locations, and if the data are routed according to the shortest path routing protocol, the problem reduces to a form of a set-covering problem in which sensor locations satisfying the coverage requirements of the field are sought. Therefore, it can be said the problem is an extremely difficult one, since the set-covering problem, which is a very simplified version of this problem, is NP-complete. As a consequence, there is almost no hope for finding a solution method that solves CASD instances in polynomial time. However, as seen in the numerical results section, the proposed solution method is able to find very good networks in a tolerable amount of computation time for instances with realistic sizes.

From the discussion above, it can be understood that it is impossible to obtain exact solutions of CASD formulation, even for small and/or moderate-sized instances, due to the high number of binary variables. This observation suggests decreasing the number of binary variables by setting the values of selected binary variables a priori before solving the model from scratch. All the variables except one (p_{ir} binary variable) have t indices. Therefore, a reduction in the number of periods has the potential to reduce the number of variables. If the number of periods is set to a small integer like 1 or 2, then the solution of the formulation becomes much easier, but the solution quality worsens. On the other hand, assigning a large number to the number of periods makes the solution of the formulation difficult. This observation suggests starting the number of periods from a low number to easily solve and obtain a solution probably having a low quality, and increasing the number of periods one-by-one to obtain solutions having better quality. The number of sinks can also be limited and increased one by one in a similar manner. This solution strategy is called the period sink shift heuristic (PSSH). If the number of periods is limited to F, only the variables belonging to the period F or the earlier periods ($w_t, z_{lt}, q_{irt}, x_{irjst}, y_{irlt}$, and a_{irt} for $t \leq F$) are allowed to take nonzero values, while variables belonging to later periods ($w_t, z_{lt}, q_{irt}, x_{irjst}, y_{irlt}$, and a_{irt} for $t > F$) are set to 0. After contracting the values of the variables in that manner, the commercial solver Gurobi can be run for a limited time. It is expected that the solver is capable of producing optimal or near-optimal solutions, as the size of the remaining problem after setting the variable values is relatively small. After finding a feasible solution for a value of F, the number of sinks can be increased by one and the solver can be rerun. The procedure can be repeated until no more improvement in the objective function is observed after an increase in the allowable number of sinks. After the convergence, the number of periods can be increased by one and the procedure can be repeated for the number of periods at the outer loop this time. Increasing the allowable number of sinks or number of periods provides a more flexible optimization framework to the commercial solver by letting it determine the values of a larger number of variables. Therefore, an increase in the network lifetime is expected as a response to the increase in the number of sinks or periods. However, the size of the model also increases with the number of sinks and periods, meaning that finding high-quality solutions will be more difficult. Fortunately, the solution of the restricted model where the number of periods is $F+1$ can be accelerated by making use of the solution of the restricted model with F number of periods. The algorithm terminates when the outer loop converges to a network lifetime. Pseudocode of the PSSH algorithm is given below.

A flowchart of PSSH is provided in Figure 2 in order to have a clearer understanding of the algorithm.

PSSH is a simple and practical method that is able to produce networks with long lifetimes, as pointed out in the next section.

4. Numerical results

In this section, the selection process of parameter values of CASD formulation is explained. Then the success of PSSH is indicated by comparing the networks found by PSSH and the commercial solver Gurobi on a large set of numerical instances.

4.1. Generation of test problem instances

Three different instances were randomly generated for each of the six sets of sensor location candidates (40, 60, 80, 100, 150, and 200 locations). It was assumed that the sensor field possessed a grid-like structure in which all horizontal and vertical distances were set to 15 m. Size of the grid was chosen so that the shape was as

PSSH:

Initialization: Set F= 1, let allowable number of sinks P= 1, DIFF1 = 100, initiate ε_1, ε_2 as small positive precision values, set the first objective function value $\lambda_0 = 0$

While (DIFF1 $> \varepsilon_1$)

  Set DIFF2 = 100, $\beta_P = \lambda_{F-,1}$

  While (DIFF2 $> \varepsilon_2$)

    Solve CASD formulation for F and P values for a limited time by making use of the last solved restricted model solution

    Let β_{P+1} be the objective function value of the restricted model, let DIFF2 = $\beta_{P+1} - \beta_P$, let $P = P+1$

  End While

Set $\lambda_{F+1} = \beta_P$, let DIFF1 = $\lambda_{F+1} - \lambda_F$ and $F = F+1$

End While

Report λ_F and the related variables

close as possible to square. For instance, the size of the grids corresponding to 40 and 100 candidate sensor locations were, respectively, 5 × 8 and 10 × 10. This grid is referred to as the sensor grid in the sequel. It was also assumed that candidate sink locations also possessed a grid structure (the sink grid), which was nested within the sensor grid. The number of candidate sink locations was assumed to be equal to the half of the number of candidate sensor locations. The four corner points of the sink grid were selected as the center points of the four outermost square cells of the sensor grid. As a result of this selection mechanism, the horizontal and vertical distances between two neighboring candidate sink locations were equal to $15\times(n_1-2)/(m_1-1)$ and 15 $\times(n_2-2)/(m_2-1)$, respectively, where the sensor grid is of size $n_1 \times n_2$ and the sink grid is of size $m_1 \times m_2$.

It was assumed that there are two types of sensors having, respectively, 15 and 22 m sensing ranges and 50 and 80 m communicating ranges. Battery energies of the sensors were selected as e_1 = 100 J and e_2 = 200 J, respectively. Note that these energy values are generic and were chosen randomly solely for showing the effectiveness of the PSSH. Any real WSN designer may easily run PSSH with the actual battery energies of the sensors of their application. Both of the sensor types were assumed to produce the same amount of data per unit of time, h_r = 4096 bits per h. Deployment cost of sensors of type 1 was distributed uniformly between 1 and 10., i.e. $f_{i1} \sim$ D(1,10) for $i \in S$, while deployment of the second type sensors is more expensive and the extra cost of deploying a type 2 sensor was distributed uniformly between 0 and 5, i.e. $f_{i2} = f_{i1}$ + D(0,5) for $i \in S$. The sensor deployment budget is the weighed sum of all sensor deployment costs and all the weights were selected as 0.5, implying that $B = \sum_{i\in S} 0.5 \times (f_{i1}+f_{i2})$. Moreover, it was assumed that the set of coverage points and the set of candidate sensor locations coincided. The requirements of the coverage points were selected randomly as 1 or 2.

Finally, it was assumed that both types of sensors shared the same energy usage characteristics. Energy spent for transmission is a function of a constant and the square of the transmission distance. In other words, if d_{ij} is the distance between candidate sensor locations i and j, the energy spent by a sensor located at i to transmit a bit of data to a sensor located at j is given by $c^t_{ij}=\delta_1+\delta_2 d^2_{ij}$, where δ_1 and δ_2 are taken as 50 μJ per bit and 100 nJ per bit × m^2, respectively. Energy spent by the receiver sensor per bit of data is taken as c^r = 50 μJ per bit, while energy spent for sensing per unit bit of data per unit time is taken as c^s = 50 nJ. For determination of parameter values, [29] and [22] were used.

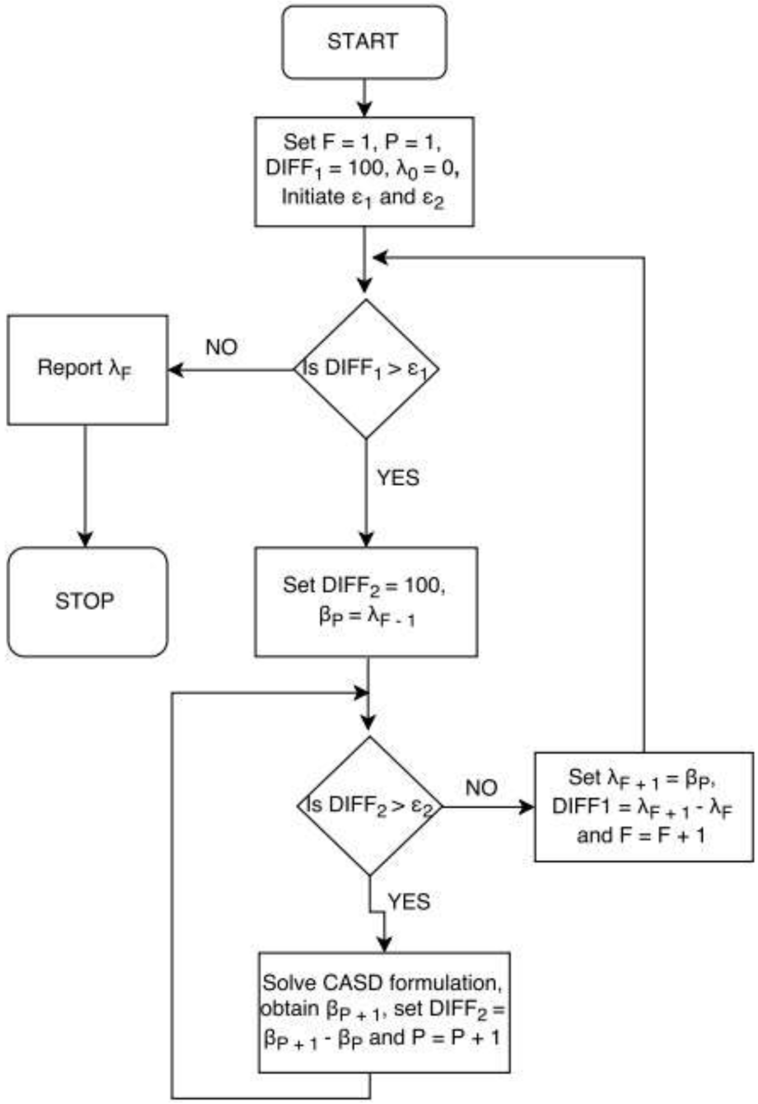

Figure 2. Flowchart of PSSH.

4.2. Performance of PSSH

In this section, the network lifetimes produced by PSSH and the state-of-the-art commercial MILP solver Gurobi (URL: http://www.gurobi.com) under the above mentioned numerical instances are compared. Note that the well-known commercial MILP-solver Cplex (URL: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer) performed poorly in instances of a formulation provided in [25], which is similar to CASD, while Gurobi produced decent networks. Hence, Gurobi was used in order to produce better networks. Both methods were coded and run in C# language in Visual Studio using a single Intel Xeon 5460 core with 28 GB of RAM. Three hours of computation times were given for each instance for both methods. If Gurobi found the optimal solution or if the PSSH heuristic converged to a solution in less than 3 h, they immediately reported the best solution they found and started to work on the next instance. The network lifetimes found by Gurobi and PSSH are given in the Table. The first column of the table contains the problem size, while columns 2 and 3 include the network lifetimes found by Gurobi and PSSH, respectively (denoted by z_H and Z_{PSSH}, respectively). The fourth column includes the percent deviation between the Gurobi and PSSH network lifetimes, calculated using the formula $100 \times \frac{(z_{PSSH} - z_G)}{z_{PSSH}}$. Finally, columns 5 and 6 include the time spent by Gurobi and PSSH (denoted by T_G and T_{PSSH}, respectively).

Table. Network lifetimes found by Gurobi and PSSH.

$\lvert S\rvert$, $\lvert L\rvert$	z_G	z_{PSSH}	% Deviation	T_G	T_{PSSH}
(40, 20)	139.61	441.93	68.41	10,800.60	618.52
(40, 20)	316.21	488.28	35.24	10,800.05	24.51
(40, 20)	129.45	418.90	69.10	10,800.10	348.93
(60, 30)	125.11	224.24	44.21	10,800.07	353.45
(60, 30)	132.83	488.28	72.80	10,800.08	792.38
(60, 30)	119.45	282.56	57.72	10,801.81	1139.05
(80, 40)	115.08	187.11	38.50	10,800.20	398.83
(80, 40)	0.00	148.51	100.00	10,800.09	132.63
(80, 40)	114.41	176.53	35.19	10,800.17	677.15
(100, 50)	114.25	202.58	43.60	10,800.17	449.98
(100, 50)	0.00	488.28	100.00	108,00.15	2859.34
(100, 50)	121.38	225.99	46.29	10,800.19	3924.02
(150, 75)	0.00	100.97	100.00	10,800.06	510.69
(150, 75)	92.55	244.14	62.09	10,800.29	236.40
(150, 75)	0.00	40.63	100.00	10,800.11	354.33
(200, 100)	0.00	160.08	100.00	10,800.20	5660.37
(200, 100)	0.00	138.71	100.00	10,800.13	3261.20
(200, 100)	0.00	77.14	100.00	10,800.28	2055.14

As can be understood from the Table, the network lifetimes found by PSSH are larger than those found by the commercial solver Gurobi for all instances. Moreover, Gurobi was unable to find a network with positive lifetime for several instances, while PSSH was able to find positive lifetimes for all of the instances. Finally, it can be seen by comparing last two columns of the table that Gurobi used 3 h in full for all instances, while PSSH used at most 2 h for all instances and around 1 h for most of the instances. In order to have a clearer picture of the relative superiority of PSSH to Gurobi, the average values of the network lifetimes found by Gurobi and PSSH, the percent deviations between them, and the computation times used by Gurobi and PSSH are calculated for each set of instances. These values are pictured against the problem size (the number of candidate sensor locations) in Figures 3–5. Finally, the energy characteristics of Gurobi and PSSH are given in Figures 6 and 7.

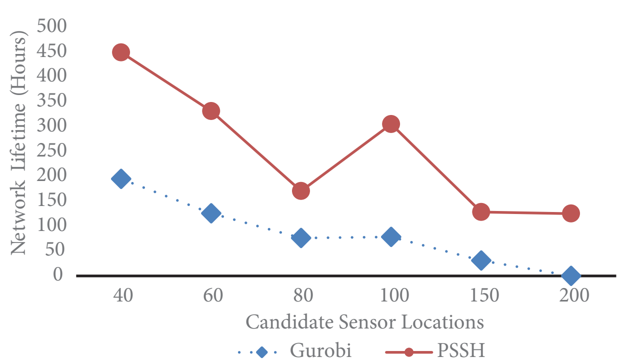

Figure 3. Gurobi and PSSH: Network lifetimes.

As seen in Figure 3, PSSH performed better than Gurobi in all instances, implying that PSSH is a more reliable solution method for CASD formulation than Gurobi. One may also infer from Figure 3 that network lifetimes found by both PSSH and Gurobi decrease as the size of the problem size increases. This decrease in the network lifetime is an expected result, since obtaining feasible solutions becomes more difficult for both methods as the problem size gets larger. Moreover, increasing the network size will probably increase the number of sensors having relatively critical levels of energy as well, and in such a case it would be difficult to help all the critical sensors with the limited number of sinks present at hand.

Percent deviations between PSSH and Gurobi results depicted in Figure 4 also confirm the conclusions drawn from Figure 3. All the percent deviations are above zero, implying that the lifetimes found by PSSH are larger than those found by Gurobi. The relative superiority of PSSH to Gurobi becomes clearer as the problem size increases.

Average computation times for the test instances are given in Figure 5. Gurobi used the entire 3 h for all instances, while PSSH used only 1 h for most of the instances. Computation time of PSSH exceeded 1 h and reached around 1.5 h only for test instances with 200 candidate sensor locations.

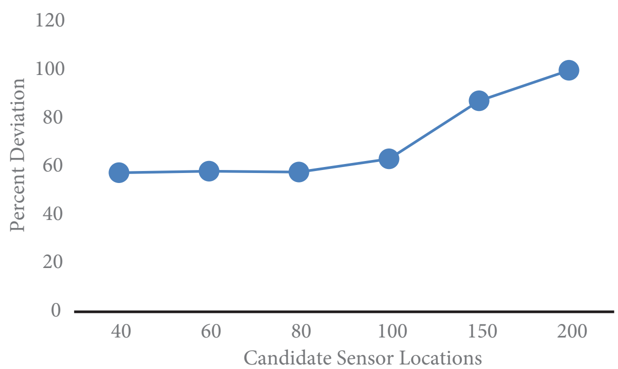

Figure 4. Gurobi and PSSH: Percent deviations.

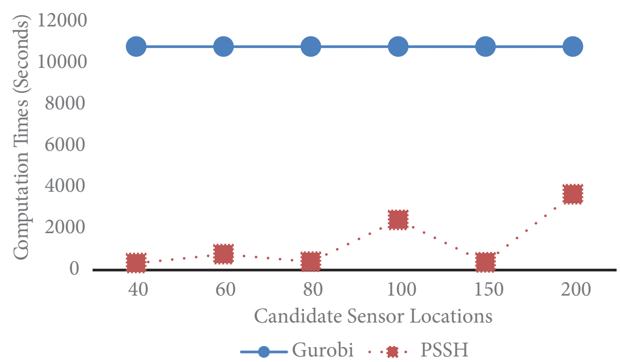

Figure 5. Gurobi and PSSH: Computation times.

Averages of the total energy spent by all sensors for varying candidate sensor locations are presented in Figure 6. Total sensor energy used was greater for PSSH than for Gurobi for all instances. At first, this result may seem contradictory. However, since networks found by PSSH lived longer than those found by Gurobi, it is expected that more energy was spent by the sensors as well. Moreover, energy loads of the network sensors found by PSSH were distributed more evenly than in the networks found by Gurobi. This implies that more sensors were able to deplete their energies during the network lifetime in networks found by PSSH than in networks found by Gurobi, which resulted in higher energy usage in the PSSH networks. The average residual energies of the sensors at the end of the lifetime are given in Figure 7 in order to have a clearer sense of the energy usage characteristics.

The average residual energies were lower for PSSH networks than Gurobi networks for all instances, meaning that the energy load of the sensors was evenly distributed for PSSH networks.

5. Discussion and conclusions

In this study, a MILP is developed in which the cost of the sinks is taken into consideration. The formulation also integrates four WSN design issues: sensor location, sensor activity scheduling, sink routing, and data routing from sensors to the sinks. A heuristic solution procedure, PSSH, is developed. In each iteration of PSSH, a limited formulation, in which the number of sinks and periods is set to small values, is solved for an allowed computation time and the number of allowed sinks and periods are increased in nested loops until no

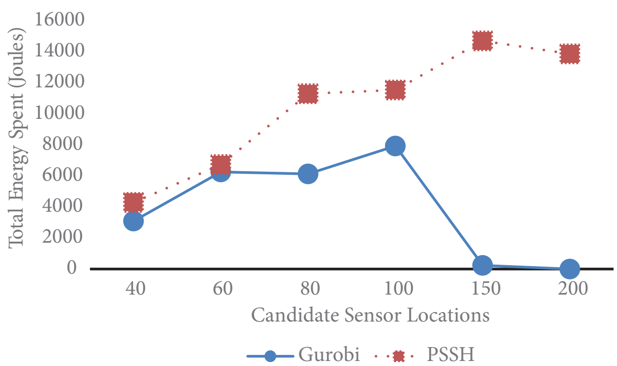

Figure 6. Gurobi and PSSH: Total energy spent.

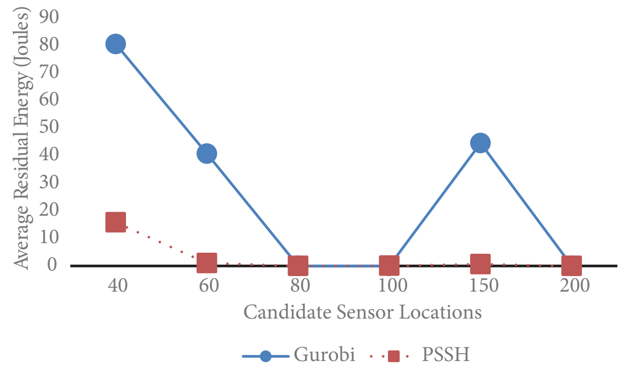

Figure 7. Gurobi and PSSH: Average residual energies.

improvement is observed in the network lifetime. The solution procedure is accelerated by using the solution found in the previous iteration as the starting point of the current iteration. Network lifetimes found by PSSH are compared with those found by the commercial solver Gurobi for an extensive number of problem instances. Consequently, it can be said that PSSH produces better networks and in less computation time than Gurobi.

This work can be extended in several ways. First of all, networks produced by PSSH can be tested with WSN simulators like TOSSIM [30] and OMNeT++ [31]. By testing the networks with a simulator, it becomes possible to assess the performance of PSSH for several WSN performance metrics other than the network lifetime, such as congestion rate, message latency rate, and throughput rate. Moreover, verification or correction of the parameter values used in the MILP formulation becomes possible after testing the networks with a simulator. Finally, different solution strategies can be developed and compared with PSSH. For instance, adapting the pioneering LEACH algorithm of [29] to include activity schedule of the sensors and variable period lengths and comparing the networks found by PSSH and LEACH adaptation would be interesting, whose undertaking is planned as a future work.

References

[1] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. Comput Netw 2008; 52: 2292-2330.

[2] Popa L, Rostamizadeh A, Karp R, Papadimitriou C, Stoica, I. Balancing traffic load in wireless networks with curveball routing. In: International Symposium on Mobile Ad Hoc Networking and Computing; 9–14 September 2007; Montréal, Québec, Canada. ACM. pp. 170-179.

[3] Li J, Mohapatra P. Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. Pervas Mobile Comput 2007; 3: 233-254.

[4] Wu X, Chen G, Das SK. Avoiding energy holes in wireless sensor networks with nonuniform node distribution. IEEE T Parall Distr 2008; 19: 710-720.

[5] Basagni S, Carosi A, Melachrinoudis E, Petrioli C, Wang ZM. Controlled sink mobility for prolonging wireless sensor networks lifetime. Wirel Netw 2008; 14: 831-858.

[6] Altınel İK, Aras N, Güney E, Ersoy C. Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks. Comput Netw 2008; 52: 2419-2431.

[7] Wang ZM, Basagni S, Melachrinoudis E, Petrioli C. Exploiting sink mobility for maximizing sensor networks lifetime. In: 38th Annual Hawaii International Conference on System Sciences; 3–6 January, 2005; Big Island, HI, USA. IEEE. p. 287a.

[8] Basagni S, Carosi A, Petrioli C, Phillips CA. Coordinated and controlled mobility of multiple sinks for maximizing the lifetime of wireless sensor networks. Wirel Netw 2011; 17: 759-778.

[9] Keskin ME, Altınel İK, Aras N, Ersoy C. Lifetime maximization in wireless sensor networks using a mobile sink with nonzero traveling time. Comput J 2011; 54: 1987-1999.

[10] Gandham SR, Dawande M, Prakash R, Venkatesan S. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In: IEEE Global Telecommunications Conference; 1–5 December 2003; San Francisco, CA, USA. IEEE. pp. 377-381.

[11] Azad AP, Chockalingam A. Mobile base stations placement and energy aware routing in wireless sensor networks. In: IEEE Wireless Communications and Networking Conference; 3–6 April 2006; Las Vegas, NV, USA. IEEE. pp. 264-269.

[12] Alsalih W, Akl S, Hassanein, H. Placement of multiple mobile base stations in wireless sensor networks. In: IEEE International Symposium on Signal Processing and Information Technology; 15–18 December 2007; Cairo, Egypt. IEEE. pp. 229-233.

[13] Jun L, Hubaux JP. Joint mobility and routing for lifetime elongation in wireless sensor networks. In: Annual Joint Conference of the IEEE Computer and Communications Societies; 13–17 March 2007; Miami, FL, USA. IEEE. pp. 1735-1746.

[14] Papadimitriou I, Georgiadis L. Maximum lifetime routing to mobile sink in wireless sensor networks. In: 15–17 September 2005; Split, Marina Frapa, Croatia. IEEE. pp. 1-5.

[15] Gatzianas M, Georgiadis L. A distributed algorithm for maximum lifetime routing in sensor networks with mobile sink. IEEE T Wirel Commun 2008; 7: 984-994.

[16] Yun Y, Xia Y. Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications. IEEE T Mobile Comput 2010; 9: 1308-1318.

[17] Yun Y, Xia Y, Behdani B, Smith JC. Distributed algorithm for lifetime maximization in a delay-tolerant wireless sensor network with a mobile sink. IEEE T Mobile Comput 2013; 12: 1920-1930.

[18] Behdani B, Yun YS, Smith JC, Xia Y. Decomposition algorithms for maximizing the lifetime of wireless sensor networks with mobile sinks. Comput Oper Res 2012; 39: 1054-1061.

[19] Güney E, Aras N, Altınel İK, Ersoy C. Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks. Comput Netw 2010; 54: 1805-1822.

[20] Luo J, Hubaux JP. Joint sink mobility and routing to maximize the lifetime of wireless sensor networks: the case of constrained mobility. IEEE ACM T Network 2010; 18: 871-884.

[21] Güney E, Aras N, Altınel İK, Ersoy C. Efficient solution techniques for the integrated coverage, sink location and routing problem in wireless sensor networks. Comput Netw 2012; 39: 1530-1539.

[22] Türkoğulları YB, Aras N, Altınel İK, Ersoy C. Optimal placement, scheduling, and routing to maximize lifetime in sensor networks. J Oper Res Soc 2010; 61: 1000-1012.

[23] Türkoğulları YB, Aras N, Altınel İK, Ersoy C. A column generation based heuristic for sensor placement, activity scheduling and data routing in wireless sensor networks. Eur J Oper Res 2010; 207: 1014-1026.

[24] Türkoğulları YB, Aras N, Altınel İK, Ersoy C. An efficient heuristic for placement, scheduling and routing in wireless sensor networks. Ad Hoc Netw 2010; 8: 654-667.

[25] Keskin ME, Altınel İK, Aras N, Ersoy C. Wireless sensor network lifetime maximization by optimal sensor deployment, activity scheduling, data routing and sink mobility. Ad Hoc Netw 2014; 17: 18-36.

[26] Keskin ME, Altınel İK, Aras N, Ersoy C. Wireless sensor network design by lifetime maximisation: an empirical evaluation of integrating major design issues and sink mobility. Int J Sens Netw 2016; 20: 131-146.

[27] Keskin ME. A column generation heuristic for optimal wireless sensor network design with mobile sinks. Eur J Oper Res 2017; 260: 291-304.

[28] Khalily M, Shamsi M, Nadjafi-Arani MJ. A convex optimization model for topology control in network-coding-based wireless sensor networks. Ad Hoc Netw 2017; 59: 1-11.

[29] Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: 33rd Annual Hawaii International Conference on System Sciences; 4–7 January, 2000; Big Island, HI, USA. IEEE. pp. 1-10.

[30] Levis P, Lee N, Welsh M, Culler D. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In: International Conference on Embedded Networked Sensor Systems; 5–7 November 2003; Los Angeles, CA, USA. ACM. pp. 126-137.

[31] Varga A, Hornig R. An overview of the OMNeT++ simulation environment. In: International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops; 3–7 March 2008; Marseilles, France. Brussels, Belgium: ICST. pp. 60-70.