COVERAGE ENHANCEMENT OF AVERAGE DISTANCE BASED SELF-RELOCATION ALGORITHM USING AUGMENTED LAGRANGE OPTIMIZATION

Shivani Dadwal and T. S. Panag

Deptt of Electronics and Communication Engineering, BBSBEC, Fatehgarh Sahib, India

ABSTRACT

Mobile robots with sensors installed on them are used in wireless sensor networks to generate information about the area. These mobile robotic sensors have to relocate themselves after initial location in the field to gain maximum coverage The average distance based algorithm for relocation process of mobile sensors does not require any GPS system for tracking the robotic sensors, thus avoiding cost, but increasing energy consumption. Augmented Lagrange method is introduced in average distance based algorithm to simplify the boundary resolutions and increase the coverage area of sensors located in the field.

KEYWORDS

Average distance based self-relocation, modified average distance self-relocation.

1. INTRODUCTION

A lot of work has been done on sensor network as it has a wide application in various fields. One of its major application is in military area wherein surveillance of enemy areas has to be done. Earlier static sensors were deployed for detecting targets in a particular area. This reduced the coverage area of sensors. With advances in mobile robotics, now sensors can be carried by robotic structures. All they need is a direction which is administered by an algorithm. Execution of these algorithm leads to deployment of sensors in an area in a manner that maximum area is covered and all the targets are well detected. Even civilian application of such mobile sensors exists in property and homeland security [1, 2].

For military purposes, the sensors are mostly dropped from air, which leads to a random deployment of sensors. These sensors have to reposition themselves. This issue has been covered in our paper. Optimization of coverage has been an active topic in sensor networks as along with coverage there are many other issues like minimum energy consumption and maximum lifetime which haveto be kept in mind. Optimization of static sensor networks has been done using Genetic Algorithms which helps in locating sensors to their best position for maximum coverage and saves energy resulting in increased lifetime [3,4,5,6]. Other than this a few distributed and centralized algorithms have been introduced in an effort to modify and improve sensor networking [7]. The Genetic algorithm for optimization cannot be used for distributed algorithms as they require a central operating system to control the sensor positioning. Moreover Genetic algorithm can be used only where the area is well known.

Many algorithms have been developed for placing the sensors evenly in a field, out of which potential field algorithm is one [8]. In this algorithm, a potential field is generated by the sensors

DOI : 10.5121/ijngn.2015.7302

and they move accordingly. Another algorithm is the virtual force algorithm [9] in which the sensors move as per the attractive and repulsive forces generated by the sensors depending on the distances between them and attain their final positions. Some other methods namely, density control method [10] and fluid model based method [11] have also been introduced. There is another algorithm that repairs the coverage by finding the terminated sensors and moves these sensors to uncovered areas [12]. These algorithms listed above are applicable for distributed network where there is no central node and each sensor decides its own path. For this kind of network, a GPS system is must in order to track down the position of sensors after and before relocation. GPS is global positioning system that consumes a lot of power and is costly. Every time the sensor move during the execution of their respective algorithms, they send information to the GPS system which again consumes power.

Some methods were introduced to decrease the power consumption of sensor out of which one was to reduce the sensing range of sensor [13]. This was done on non-mobile sensors. It saves energy and increases the lifetime of sensors. The average distance based relocation process does not use any GPS system and hence cuts the energy consumption and cost as well [14]. But since there is no GPS used, the sensors consume extra energy to find their best final position which also increases the number of iterations to find the final coverage. Hence, optimization of average distance based self-relocation process is introduced to facilitate the sensors to find their final positions in less time which would also reduce energy consumption. This is done using augmented lagrangian optimization method [19]. In the next section, average distance based self-relocation algorithm has been discussed.

2. AVERAGE DISTANCE BASED SELF RELOCATION PROCEESS

2.1. Assumptions

Following are a few assumptions that we consider for the algorithm:

- Each sensor has a sensing area in the form of a circle, having radius, r. The probability of covering this area is 1.
- The area A, where the sensors are randomly deployed is known by the sensors approximately
- If the sensors come within their sensing range Rc, the strength of the signals transmitted by each sensor can be measured by the other.
- All sensors have certain range of communication and have a transmission power.
- Sensors have the ability to move as per the coordinates given after execution of algorithm.
- Sensors can detect obstacles in the field.
- Sensors that meet an obstacle in the field has its movement blocked and cannot communicate with the rest of sensors.

2.2. Framework

The main aim of this algorithm is designing a distributed algorithm which has a self- relocation capability to optimize the coverage area of field using less energy. It this algorithm, the distances between the sensors have to be known in order to relocate the sensors. The sensors transmit signal in the field once they are randomly deployed. These signals are intercepted by the sensors that come within the reach of .sensor areas of other sensors. The received signal strength is measured and corresponding distance is known.

Firstly, a “hello” signal is transmitted by the sensors which gives the signal strength to all the sensors, near or far, lying in sensor range. The distance corresponding to this signal strength is calculated by the sensors. Taking these distance information into account, the sensors move towards or away to each other and relocate themselves. Also, any obstacle coming in way has to be avoided by the sensors.

3. METHODOLOGY

3.1. Ideal Deployment

The ideal deployment is achieved when there are no spaces between the sensors. Such a condition is possible when the distance between the sensors is $\sqrt{3}r$ [15]. This is shown in Figure 1.Since it is not possible to achieve an ideal condition, in this algorithm, we try to achieve a near to ideal condition by placing sensors close. But as the field may have a deformity of obstacles, and various other factors, such a condition is difficult to achieve.

3.2. Calculation of Threshold

The total number of sensors in the field and the sensing field area are used to calculate the threshold distance dth and sensing radius that are near to ideal deployment. The threshold distance dth decides the sensor movement.

Let the total area of the sensor field be A. As shown in Figure 1, assume that each sensor has an effective area of coverage, E as in [14]. Let the total number of sensors deployed in the field be N. Effective coverage of each sensor is given as:

$$E = \frac{A}{N} \qquad (1)$$

Also, E is a hexagonal area. Area of a hexagon is given as:

$$E = \frac{3\sqrt{3}}{2}.r^2 \qquad (2)$$

The threshold distance and sensing radius are calculated by the following equations:

$$r = \sqrt{\frac{2}{3\sqrt{3}}.\frac{A}{N}} \qquad (3)$$

$$d_{th} = \sqrt{3}r \qquad (4)$$

The effective coverage, E should be larger than this value, as the sensors which lie close to obstacles or to the edges will have less coverage. Hence, the threshold distance and sensor radius are increased by 15%.

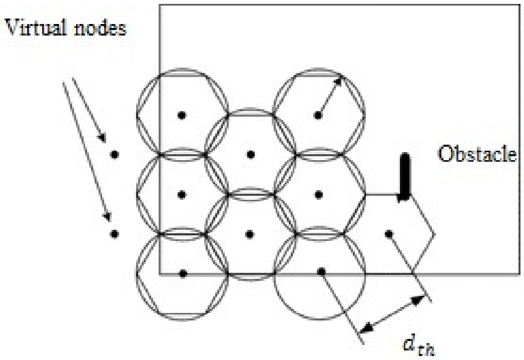

Figure 1. : Ideal coverage

3.3. Virtual Nodes

The algorithm considers that there exist virtual nodes at the boundary of the field. This is done to make deployment easy for the sensors lying close to the edges. Such virtual nodes do not actually exist. They are just used to avoid the sensors from getting closer to the edges. As shown in Figure 1, virtual nodes are considered along the boundary. No virtual nodes are required in the optimization technique used in this work.

3.4. Movement Standards

The sensors relocate themselves by adjusting the distances between themselves. They either move far or get closer to each other. No sensor has information about the direction of the other sensor. The criteria in which the sensors move is described as below:
Standard 1: If there is at least one sensor in the communication range of sensor S having distance less than $0.9d_{th}$, then the sensor will move away from other sensors.

Standard 2: If the standard 1 is not met and not more than 2 sensors lie at distance less than $1.1d_{th}$from S, then the sensor needs to move closer to other sensors.

*Standard 3:*Sensor S need not move if the above two standards are not met.

A 10% margin is kept along these standards, so that the sensor is able to achieve a distance nearer to $d_{th}$from the rest of sensors.

3.5. Moving Distance

The movement of sensors is based on two standards, hence the moving distance is calculated by the following equation:

$$d_{travel} = \begin{cases} d_{th} - \frac{1}{m_1}\sum_{j=1}^{m_1} d_j & for\ standard\ 1 \\ \frac{1}{m_2}\sum_{i=1}^{m_2} d_i - d_{th} & for\ standard\ 2 \\ 0 & for\ standard\ 3 \end{cases} (5$$

In the above equation, d_j and d_i are the distances from sensor S to other sensors. In the first standard, the sensor moves only for the sensors closer than distance threshold. The total number of sensors is given by m_1. In the second standard, all the neighbouring sensors to sensor S are taken into account. The total number of sensors is m_2.

The direction of movement of the sensor is chosen randomly as the sensor is unaware of the neighbouring sensors. There can be a back and forth movement of sensors. To avoid this direction control scheme is used. As the sensors move, the difference of direction of movement is kept less than 90 degrees. Let the last direction of movement be α, then in the next movement direction has to be in between α-90 degrees to α+90 degrees. When standard 1 is executed, the sensors move away from each other and when standard 2 is executed, the sensors move nearer to each other. As the sensors are moving to the direction chosen randomly, they check after moving through a short distance if the required coverage is attained. If not, they come back to their original positions.

4. SIMULATION AND ANALYSIS

For examining the results, we consider that the sensing field is a 60 by 60 grid structure. Each grid is 1 meter apart from the other grid. Consider that the sensing range of the sensors used lies within 18 to 25 meters. Hence, the maximum sensing radius is 25 meters. The range of communication of sensors is kept almost double the maximum sensing radius i.e. 55 meters.

4.1. AnAverage Distance Based Self- relocation Algorithm Performance

Let 20 sensors be randomly deployed in the sensor field. The distance threshold and sensing radius can be calculated from equations (3) and (4). A 15% increase should also be considered as explained in Section 2.3.The following equations give the sensing radius and threshold distance of 20 sensors:

$$r = 1.15\sqrt{\frac{2}{3\sqrt{3}}.\frac{A}{N}} = 1.15\sqrt{\frac{2}{3\sqrt{3}}\frac{60\times60}{20}} = 9.57\ meters$$

$$d_{th} = \sqrt{3}r = \sqrt{3}\times 9.57 = 16.57\ meters$$

To analyse the results, three different conditions of initial sensor placement can be considered. In the first case, the sensors are all placed in the center of the sensing field such that they cover 50 by 50 meters area. In the second condition, the sensors are deployed or scattered in the whole sensing area. In the third condition, sensor are divided into 2 groups which are separately placed in the sensing field. The coverage initially is calculated for all the conditions which come out to be 55%, 50% and 61% approximately. The coverage can be calculated by the following equations:

$$R_{coverage} = \frac{A_{covered}}{A_{total}} \qquad (6)$$

Here,
$R_{coverage} = coverage\ ratio$
$A_{covered} = area\ covered\ by\ sensors\ cooperatively$
$A_{total} = sensing\ field\ area$
If we have to find out the coverage from the 100 by 100 grid structure, then the following equation can be used:

$$R_{coverage} = \frac{n}{N}(7)$$

Here,
n= number of grid points covered by sensors
N= total number of grid points.

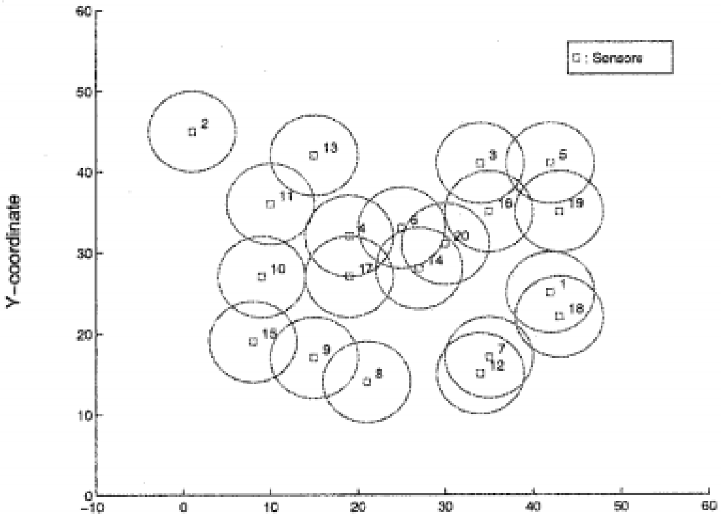

Figure 2: Random placement of sensors for scenario 2

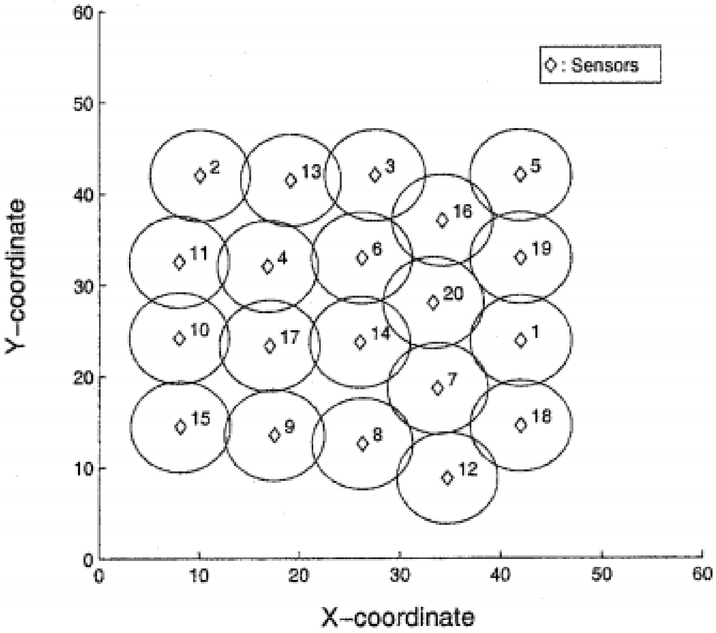

Figure 3: Final placement of sensors after execution of average distance based algorithm for scenario

As shown in figures 2, there are 20 sensors placed in a field randomly. This is the initial condition where the sensors are dropped into the field. The sensors will take positions that will not provide good coverage.

After the execution of average distance algorithm the sensors take their positions as shown in figure 3. The results are for scenario 2 in which random initial position is considered. As seen in figure 3, the sensors cover the field, leaving the border areas. This space is indicative of the factor that virtual nodes are considered out of the boundary so that the sensors do not cross the field, leaving some area uncovered. The following section will compare the results of three different cases with different initial deployments.

4.2. Coverage Analysis

The sensors in this algorithm do not have a fixed direction of movement. They move randomly and check if the required conditions are met. So, each of the three conditions stated in the above section are run 10000 times with a desired round number of 20. The execution results are shown in the Figure 4.

In figure 4, the results of average distance based algorithm are compared with virtual force algorithm. The virtual force algorithm is executed by taking its parameters into account as given in [16]. In all the three conditions, the coverage increases as the number of rounds increase. An average distance based self-relocation algorithm can achieve nearly 94% of coverage in the sensing field after 20 rounds. For a virtual force algorithm, after 20 rounds 93% coverage is achieved.

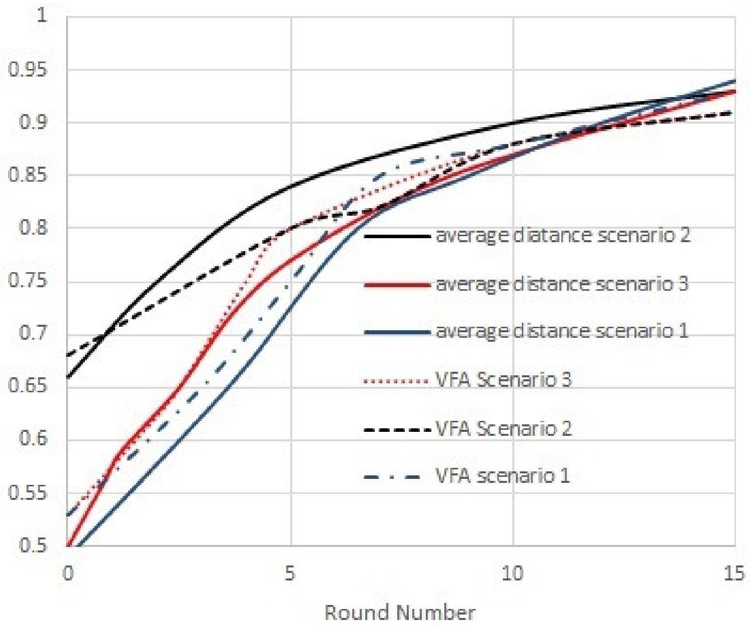

Figure 4:Simulation results of average coverage vs round number for average distance based self –relocation algorithm

Thus, both the algorithm lead to almost same coverage of sensing field by the sensors. The major difference is that the average distance based algorithm does not require a GPS hardware, reducing the cost of the sensor network. It can hence be used in those areas where GPS cannot operate, like in under water systems.

4.3. Energy Analysis

An energy analysis model has been discussed in [17]. This model has been used for mobile robots as per which energy consumed by a robot to move 1 meter is equal to 9.34 Joules if it is moving at a speed of 0.08 m/s constantly. The amount of energy used by robots to turn by 90 degrees is 2.35 Joules. Both travelling and turning of mobile robots is considered to be at a constant speed. By doing so, we can plot a linear graph between energy consumed by mobile robot and coverage. As per the above discussion, we can divide energy consumption into two following parts:

- *Energy used while travelling:* If the mobile robot keeps moving at a constant rate in one direction, then the energy consumed for one single sensor node in a single round is given as:

$$E_{travel} = d_{travel} \times 9.34(Joules)$$

- *Energy used in direction changing:* As discussed in the relocation algorithm, the sensor move randomly in some direction and checks its position by recalculating the signal strength. After this, it decides whether to move back to original position or not. Hence, during this process, the energy used by the sensor to turn in some direction is given as:

$$E_{turn} = \begin{cases} (A_{diff}/90) \times 2.35(Joules) & Keep\ moivng \\ (360/90) \times 2.35(Joules) & Turning\ back \end{cases}$$

In the above equation, A_{diff} is the difference of direction between the previous and later direction of sensor movement. If the new position does not satisfy the coverage requirement of sensor, then it gets back to position where it started by moving at 360 degrees, as shown in the above equation.

To plot a graph, virtual force algorithm has also been considered. Its energy consumption is calculated. In the virtual force algorithm, the above two energy consumption are added to the energy used by the GPS system in locating the sensor nodes and in exchange of information between sensor and GPS. The GPS chip uses 198 MW as in [18]. As the sensors are moving constantly at 0.08m/s, the GPS consumes energy per meter given by following equation:

$$0.198 \times (1/0.08) = 2.475\, Joules/meter$$

As given in the Figure 5, virtual force algorithm will consume lesser energy as compared to average distance based algorithm. The graph is plotted between average energy consumption and average coverage for both the algorithms. The VFA uses lesser energy as it is GPS enabled, making it easier for the sensor to redeploy themselves faster than the sensors in average distance based algorithm. In VFA sensors take less time to redeploy as they know their positions corresponding to other sensors and also know the locations of rest of the sensors in the field.

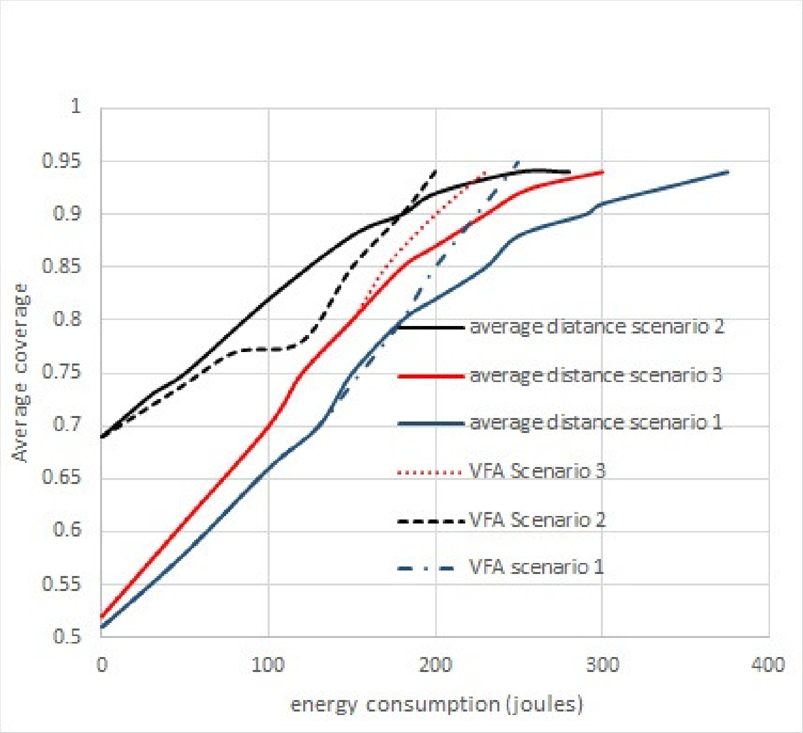

Figure 5:Simulation results for average coverage vs energy consumption

5. OPTIMIZATION USING AUGMENTED LANGRANGIAN METHOD

Augmented Lagrangian methods are a certain class of algorithms for solving constrained optimization problems. They have similarities to penalty methods in that they replace a constrained optimization problem by a series of unconstrained problems and add a penalty term to the objective; the difference is that the augmented Lagrangian method adds yet another term, designed to mimic a Lagrange multiplier. The augmented Lagrangian is not the same as the method of Lagrange multipliers. Viewed differently, the unconstrained objective is the Lagrangian of the constrained problem, with an additional penalty term (the augmentation).

The method was originally known as the method of multipliers, and was studied much in the 1970 and 1980s as a good alternative to penalty methods. It was first discussed by Magnus Hestenes [20] and by Powell [21]. The method was studied by R. Tyrrell Rockafellar in relation to Fenchel duality, particularly in relation to proximal-point methods, Moreau–Yosida regularization, and maximal monotone operators: These methods were used in structural optimization. The method was also studied by Dimitri Bertsekas, notably in his book together with extensions involving nonquadratic regularization functions, such as entropic regularization, which gives rise to the "exponential method of multipliers," a method that handles inequality constraints with a twice differentiable augmented Lagrangian function.

Since the 1970s, sequential quadratic programming (SQP) and interior point methods (IPM) have had increasing attention, in part because they more easily use sparse matrix subroutines from numerical software libraries, and in part because IPMs have proven complexity results via the theory of self-concordant functions. The augmented Lagrangian method was rejuvenated by the optimization systems LANCELOT and AMPL, which allowed sparse matrix techniques to be used on seemingly dense but "partially separable" problems. The method is still useful for some problems. Around 2007, there was a resurgence of augmented Lagrangian methods in fields such as total-variation denoising and compressed sensing. In particular, a variant of the standard augmented Lagrangian method that uses partial updates (similar to the Gauss-Seidel method for solving linear equations) known as the alternating direction method of multipliers or ADMM gained some attention.

5.1 General method

Let us say we are solving the following constrained problem:

$$\min f(\mathbf{x})$$

subject to

$$c_i(\mathbf{x}) = 0 \ \forall i \in I.$$

This problem can be solved as a series of unconstrained minimization problems. For reference, we first list the penalty method approach:

$$\min \Phi_k(\mathbf{x}) = f(\mathbf{x}) + \mu_k \sum_{i \in I} c_i(\mathbf{x})^2$$

The penalty method solves this problem, then at the next iteration it re-solves the problem using a larger value of μ_k (and using the old solution as the initial guess).

The augmented Lagrangian method uses the following unconstrained objective:

$$\min \Phi_k(\mathbf{x}) = f(\mathbf{x}) + \frac{\mu_k}{2} \sum_{i \in I} c_i(\mathbf{x})^2 - \sum_{i \in I} \lambda_i c_i(\mathbf{x})$$

and after each iteration, in addition to updating μ_k, the variable λ is also updated according to the rule

$$\lambda_i \leftarrow \lambda_i - \mu_k c_i(\mathbf{x}_k)$$

where $\mathbf{x}_k$ is the solution to the unconstrained problem at the *k*th step, i.e. $\mathbf{x}_k = \text{argmin}\Phi_k(\mathbf{x})$

The variable λ is an estimate of the Lagrange multiplier, and the accuracy of this estimate improves at every step. The major advantage of the method is that unlike the penalty method, it is not necessary to take $\mu \rightarrow \infty$ in order to solve the original constrained problem. Instead, because of the presence of the Lagrange multiplier term, μ can stay much smaller.

The method can be extended to handle inequality constraints.

5.2 Optimization on average distance algorithm

The basic drawback of average distance based algorithm is that it consumes more energy in relocating the sensors as they move back and forth many times to come to an appropriate position to get a good coverage.

Here, if an optimization technique is used to help sensors in their relocation process, they can come to final position in lesser time consuming lesser energy. In the average distance based self-relocation process, virtual nodes are considered at the boundary and outside the boundary as shown in figure 1. Using these virtual nodes, the sensors make an idea of their boundary and the area beyond which they are restricted. The sensors move back and forth in self –relocation process, during which the sensors nearer to the boundary have to recalculate its positions to stay within the boundary, which can be energy consuming. Thus, in the augmented lagrangian method, the sensors are applied a penalty function. Using this penalty function, if the sensor skips outside the boundary by certain distance d_{out}, it is made to to come inside the boundary by the same distance d_{out},as measured from the boundary. During this process, the sensor might overlap or come into the boundary of another sensor, violating the threshold distance conditions as discussed in section 2.3.

In theaugmented lagrangian optimization method, an objective function has to be considered. The minimum distance between the sensors should be d_{th}, as in average distance based relocation process. This minimum distance might change when penalty function is applied, so minimum distance between sensor i.e. threshold distance d_{th} is the objective function subject to constraints in augmented lagrangian optimization method.

Now let the boundary of the sensor field be defined by following functions.

$$g_1 > 0, g_2 \geq 0, g_3 < 0, \ldots, g_n > 0$$

Here, $g_1, g_2, g_{3,} \ldots, g_n$ are the boundary equations for sensor field and n is the number of boundaries.

In augmented lagrangian optimization method, new objective function is given as

$$f[X] - \alpha\langle d_{th}\rangle - \beta\langle g_1[X] + g_2[X] + \cdots + g_n[X]\rangle$$

$[X]$is the co-ordinate of all the sensors obtained after first iteration of average distance based relocation algorithm. The outline of the algorithm is as follows:

- The $[X]$ co-ordinates of all the sensors in the sensing field are obtained after the execution of first iteration of average distance based relocation algorithm.
- The boundary constraint violations are checked i.e. it is checked if any sensor is lying on the boundary or beyond the boundary. The constraints are checked n times which is equal to the number of sensors in the field.
- If any constraint is violated, let us say for nth sensor,$[g_1]_i \nless 0$, which means that constraint$g_1$is violated by ith sensor and the sensor is moving beyond the boundary of sensing field. Hence, the distance D_i is calculated between $g_1$and $[X, Y]_i$. X and Y are the co-ordinated of ith sensor lying outside the boundary.
- To compensate for distance and to bring the sensor back inside the field, negative of distance D_i is added in g_1 and the new co-ordinates of ith sensor are calculated.
- Now, all the co-ordinates serve as initial solution for average distance based algorithm which is run again.

- The threshold distance are again checked in the self -relocation algorithm.

Hence, this method is a modified version of average distance based self-relocation algorithm using the properties of augmented lagrangian optimization technique.

5.3 Flowchart

The flowchart describing the augumented Lagrange method applied to average distance algorithm is shown below:

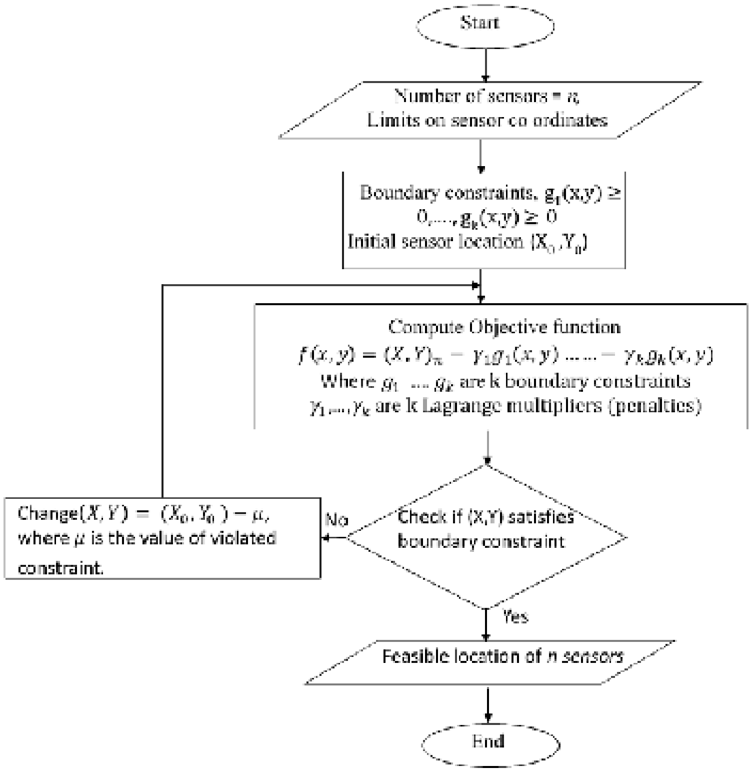
Start
Number of sensors = n,
Limits on sensor co ordinates
Boundary constraints, $g_1(x,y) \geq 0,\dots,g_k(x,y) \geq 0$
Initial sensor location (X_0, Y_0)
Compute Objective function
$f(x,y) = (X,Y)_n - \gamma_1 g_1(x,y) \dots\dots - \gamma_k g_k(x,y)$
Where $g_1 \dots g_k$ are k boundary constraints
$\gamma_1,\dots,\gamma_k$ are k Lagrange multipliers (penalties)
Check if (X,Y) satisfies boundary constraint
No
Change$(X,Y) = (X_0, Y_0) - \mu$, where μ is the value of violated constraint.
Yes
Feasible location of n sensors
End

Figure 6: Flowchart for Augumented Lagrange method

5.4 Simulation and Analysis

The analysis of lagrangian optimization technique is done by comparing its energy analysis graph with average distance based relocation algorithm[22]. The graph is given below. As it is seen from the graph, the coverageof sensors in the area has increased for all the three scenarios.

Here again 3 scenarios are considered. In the first one the sensors are deployed randomly such that they accumulate at the same place. The coverage initially is 49%. In the second condition, the sensors are deployed or scattered in the whole sensing area with initial coverage of 61%. In the third condition, sensor are divided into 2 groups which are separately placed in the sensing field having initial coverage 50%. All the cases in the optimization algorithm converge to almost 96% coverage. Thus, the coverage increases by 2% as compared to average distance based deployment but the energy consumption is same {figure 7). The energy consumed by optimization technique lies between 300 Joules to 400 Joules. Just as was in the case of average distance based algorithm.

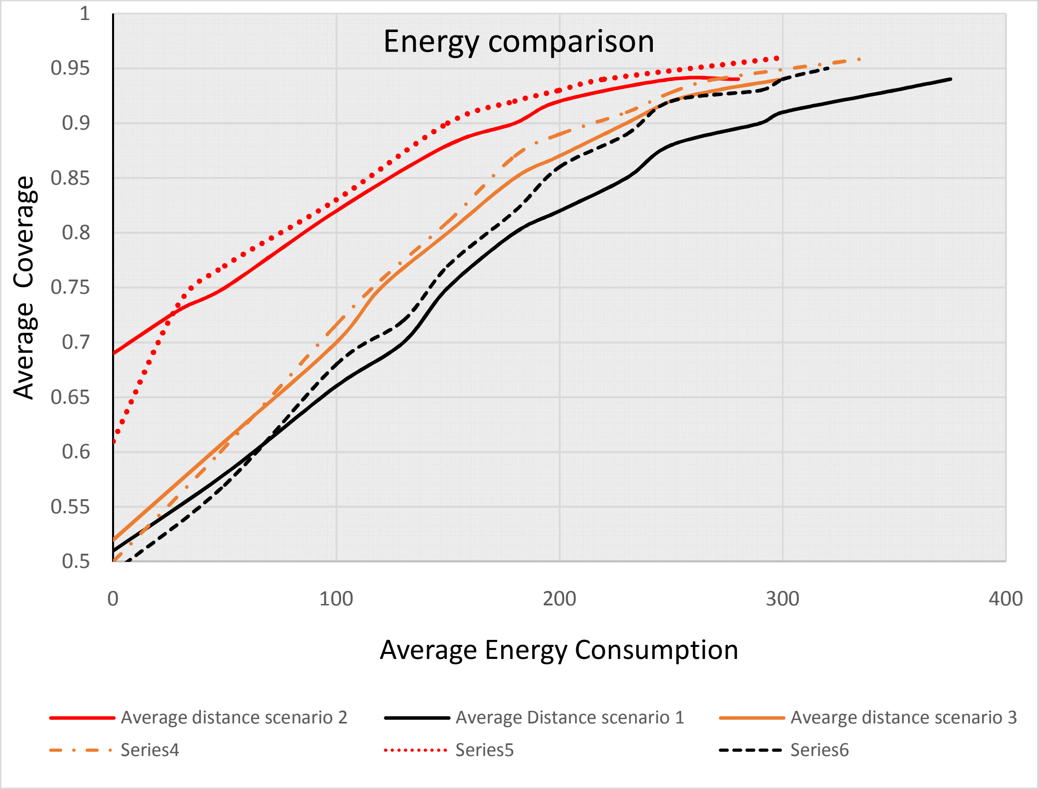

Figure 7: Graph showing energy and coverage comparisons of average distance based self-relocation, with and without optimization

6. Future Work

The biggest advantage of the relocation scheme discussed is non- requirement of a GPS hardware which cuts cost and increases the applicability of this relocation process where GPS cannot be used. The above discussion focused on optimizing the deployment of wireless sensors so the coverage and lifetime of the sensor network can be optimized. This work can be expanded further.

Algorithms have been tested in the software simulations. These simulations can be expanded to experimental works with deployment of real mobile wireless sensor networks. Programmable mobile sensor kits can be used and relocation algorithms can be tested in indoor as well as outdoor locations. Different hardware modules can be added or removed from sensor nodes for specific applications.

References

[1] V. Potdar, A. Sharif and E. Chang, “Wireless Sensor Networks: A Survey,” Advanced Information Networking and Applications Workshops, Bradford, 26-29 May 2009, pp. 636-641.

[2] Ding, W. and Marchionini, G. 1997. A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[3] J. Chen and C. Li, “Coverage Optimization Based on Improved NSGA-II in Wireless Sensor Network,” IEEE International Conference on Integration Technology (ICIT), Shenzhen, 20-24 March 2007, pp. 614-618.

[4] X. Wang, S. Wang and D. W. Bi, “Dynamic Sensor Nodes Selection Strategy for Wireless Sensor Networks,” 7th International Symposium on Communications and Information Technologies (ISCIT), Sydney, 16-19 October 2007, pp. 1137-1142.

[5] J. Weck,”Layout Optimization for a Wireless Sensor Network Using a Multi Objective Genetic Algorithm,” IEEE 59th Vehicular Technology Conference, Milan, Vol.5, 2004, pp.2466-2470.

[6] L.C. Wei, C. W. Kang and J. H. Chen, “A Force-Driven Evolutionary Approach for Multi-Objective 3D Differentiated Sensor Network Deployment,” IEEE 6th International Conference on Mobile Ad-hoc and Sensor Systems (MASS), Macau, 12-15 October 2009, pp. 983-988

[7] S. Dadwal, T.S. Panag, “Sensor Deployment Strategies in WSNs,” 3rd National Conference on Recent Advances in Electronics and Communication Technologies, Ludhiana, 21-22 March 2013, pp 412-419.

[8] A. Howard, M. Mataric and G. Sukhatme, “Mobile Sensor Network Deployment Using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem,” The 6th International Symposium on Distributed Autonomous Robotics System, Fukuoka, 25-27 June 2002, pp. 299-308.

[9] Yi. Zou and K. Chakrabarty, “Sensor deployment and Target Localization Based on Virtual Forces,” IEEE Societies Twenty Second Annual Joint Conference of the IEEE Computer and Communications (INFOCOM), San Fransisco, 1-3 April 2003, pp. 1293-1303

[10] M. R. Pac, A. M. Erkmen and I. Erkmen, “Scalable Self- Deployment of Mobile Sensor Networks: A Fluid Dynamics Approach,” Proceedings of IEEE International Conference on Intelligent Robots and Systems (RSJ), Beijing, 9-15 October 2006, pp. 1446-1451.

[11] R.-S. Chang and S.-H. Wang, “Self-Deployment by Density Control in Sensor Networks,” IEEE Transactions on Vehicular Technology, Vol. 57, No. 3, 2008, pp. 1745-755.

[12] G. Wang, G. H. Cao, T. F. Porta and W. S. Zhang, “Sensor Relocation in Mobile Sensor Networks,” IEEE Societies 24th Annual Joint Conference of the IEEE Computer and Communications, Miami, 2005, pp. 2302-2312.

[13] M. Cardei, J. Wu, M. M. Lu and M. O. Pervaiz, “Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges, “ IEEE International Conference on Wireless and Mobile Computing, Net- working and Communications, Montreal, 22-24 August 2005, pp. 438-445

[14] Y. Qu, S.V. Georgakopoulos, ”An Average Distance Based Self-Relocation and Self-Healing Algorithm for Mobile Sensor Networks,” Journal of Wireless Sensor Network, Vol. 4 No.11, pp. 257-263,2012.

[15] G. Wang, G. H. Cao and T. F. Porta, “Movement-Assisted Sensor Deployment,” IEEE Transactions on Mobile Computing, Vol. 5, No. 6, 2006, pp. 640-652.

[16] W. H. Sheng, G. Tewolde and S. Ci, “Micro Mobile Ro- bots in Active Sensor Networks: Closing the Loop,” Proceedings of IEEE International Conference on Intelligent Robots and Systems (RSJ), Beijing, 9-15 October 2006, pp. 1440-1445.

[17] Y. G. Mei, Y.-H. Lu, Y. C. Hu and C. S. G. Lee, “Energy Efficient Motion Planning for Mobile Robots,” Proceedings of IEEE International Conference on Robotics and Automation (ICRA), New Orleans, 25 April-1 May 2004, pp. 4344-4349.

[18] M. K. Stojcev, M. R. Kosanovic and L. R. Golubovic, “Power Management and Energy Harvesting Techniques for Wireless Sensor Nodes,” 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services, 7-9 October 2009, pp. 65-72.

[19] S. Singh, E. Singla, “Optimal design of robotic arms in constrained environment,” TEQIP-II sponsored National conference on ‘Latest Developments in Materials, Manufacturing and Quality Control’ MMQC, Bathinda , 18-19 March 2015.

[20] M.R. Hestenes, “Multiplier and gradient methods.” Journal of Optimization Theory and Applications, 4, 1969, pp. 303-320.

[21] M.J.D. Powell,” A method for nonlinear constraints in minimization problems,” in Optimization ed. By R. Fletcher, Academic Press, New York, NY, 1969, pp. 283-298.

[22] Dadwal, Shivani, and T. S. Panag. “Optimization of average distance based self-relocation algorithm using augmented lagrangian." Computer Science & Information Technology: 193

AUTHORS

ShivaniDadwal had completed her B.Tech degree from RIEIT, Railmajra. Now she is pursuing her M.Tech from BBSEBC, Fatehgarh Sahib.