ICS 2011
12th INFORMS Computing Society Conference

© 2011 INFORMS | ISBN 978-0-9843378-1-1
DOI 10.1287/ics.2011.0013

Sensor Deployment Through Geometric Cooperation

David C. Arney, Kristin Arney, Elisha Peterson
Department of Mathematical Sciences, United States Military Academy, West Point, New York 10996; and Johns Hopkins University Applied Physics Lab, Laurel, Maryland 20723
{david.arney@usma.edu, kristin.arney@usma.edu, elisha.peterson@jhuapl.com}

Abstract Mobile ad hoc sensor networks often need cooperation—sensors working together to achieve their common goal. How does cooperation help mobile autonomous sensors position themselves in effective locations? This paper uses mathematical simulation to study this question about sensor deployment and efficiency. The goal is to distribute sensors over a region of interest in the plane in a balanced way to ensure uniform coverage and equalized load. This paper generalizes an algorithm for accomplishing this in a distributed, cooperative, computationally efficient manner, while also allowing for a generalized notion of balance for nonhomogeneous zones and variable-capable sensors (i.e., some zones are more important than others and some sensors are more capable than others).

Keywords networks; sensors; architecture; distribution

1. Introduction

The Army's critical operational entities and networks consist not only of humans and soldiers, but also of unmanned systems (machines, computers, robots, and sensor networks). One of the fundamental ways that entities like mobile sensors cooperate with one another is by geometric movement and positioning (Arney and Peterson [1], Arney et al. [2, 3], Wang et al. [14]). For example, sensors deployed in surveillance of a region may be assigned zones of equal area to monitor, units in a military force may be assigned sectors of equal area to control, or satellites may be distributed in order to observe the entire battlespace. In such cases, the assigned areas may need to shift as the situation changes. We call these *equidistribution* scenarios. As the dynamics of the situation unfold and some entities move, withdraw, or enter the space, other entities cooperate by adjusting their geometric positions to maintain the balance of assigned areas. The sensors are assumed to operate autonomously; there is no control mechanism for positioning all sensors, and each sensor may only be able to communicate with a few nearby sensors. A natural consequence of this equidistribution process is a static sensor positioning algorithm that provides balanced area coverage for effective surveillance operations.

This paper generalizes an existing algorithm and cooperation framework for autonomous sensor movement and location within a prescribed region through geometric equidistribution in both static and dynamic situations (Arney et al. [2, 3]), and tests the generalized algorithm's performance in a simulation environment. As the algorithm runs, each sensor cooperates by moving to balance its area of responsibility, essentially moving from densely deployed areas to sparsely deployed areas in an iterative manner. The algorithm studied

in this paper was introduced for homogeneous sensors and regions in prior work (Arney et al. [2, 3]). This paper extends the results to the nonhomogeneous case. The algorithm can accommodate sensors with varying capabilities (longer range or more robust performance) and selected regions with higher priorities that need denser sensor coverage. There are many other considerations for sensor network performance, such as communication, stealth, energy consumption, individual sensor performance, obstacles, and maneuvering restrictions (Frank and Römer [7], Kang and Golay [8], Meguerdichian et al. [9], Mulligan and Ammari [10]). These performance criteria are not directly addressed in this paper, although the results could be extended to include these other criteria within the cooperation framework.

2. Geometric Equidistribution

2.1. Area Equalization

The central question of this paper is how to autonomously position n sensors in a polygonal region in a "balanced" way. The notion of balance depends on the Voronoi partition of the polygonal region determined by the sensor locations; the partition yields an assignment of a *sensor-region* to each sensor, comprising the points closer to that sensor than to any other, and an optimal algorithm positions the sensors so that each sensor has the same area to cover, which we call an *equal-area division* (Peterson [12]). In an autonomous (cooperative) *equidistribution* algorithm, the mobile sensors "sense" their situation and move autonomously in an attempt to balance the areas of their assigned sensor-regions. The algorithm determine how close the final division is to an equal-area division.

There are some classic static algorithms that try to use established shapes (triangles, polygons) to divide a region (Ben et al. [5], Snyder [13]). Other approaches use Voronoi diagrams centered around various points in the region, similar to the approach used here (Ohyama [11], Snyder [13]). Still other algorithms provide sensor locations or metrics for special geometric cases, but do not have the flexibility or utility of these new algorithms (Aurenhammer [4], Butler et al. [6], Kang and Golay [8], Meguerdichian et al. [9]).

2.2. Metrics

We examine the performance of equidistribution algorithms as follows. Initially, the sensors are dispersed at random within a prescribed polygonal region. A Voronoi diagram is used to assign each sensor a coverage area consisting of those points closer to it than to any other sensor, as in Figure 1. The algorithm is then iterated several times, simulating a dynamic scenario in which the sensors move simultaneously according to their algorithms, seeking to reduce the imbalance in assigned coverage areas. Metrics are computed to determine how successful (near equal) the sensor distribution is over time.

The following performance metrics capture average deviations of the coverage areas from the mean area; the deviations are normalized with respect to the total area. Let $i = 1, 2, \ldots, n$ denote the sensors, let $A_1, A_2, \ldots, A_n$ represent the areas of the sensor-regions, and let $\bar{A}$ represent the mean of these areas. We track three normalized measures of deviation:

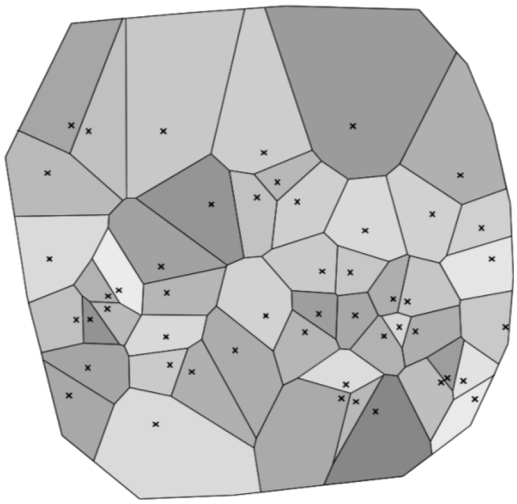
(1) The *maximum deviation* from the mean, as a ratio with mean sensor-region area: $\max_i |A_i - \bar{A}|/\bar{A}$.

(2) The *mean deviation* from the area mean, as a ratio with mean sensor-region area: $(1/n)\sum |A_i - \bar{A}|/\bar{A}$.

(3) The average of the *mean-squared deviations* as a ratio with the square of the mean sensor-region area: $(1/n)\sum |A_i - \bar{A}|^2/\bar{A}^2$.

We also track a *utility score* measuring the total deviation of all sensors from the mean. Each sensor contributes at most 1 to this metric, so that a maximum value of n corresponds to an equal-area division. In prior work, this utility metric was used to provide a quantitative

FIGURE 1. Polygonal region divided into sensor-regions based on the random locations of 50 sensors.



Note. This partition has poor utility metrics because the largest assigned area is over three times larger than the mean area.

basis for comparing the cooperative value of various algorithms (Arney and Peterson [1], Arney et al. [2, 3]).

2.3. Algorithms

The algorithms go into effect when the sensors determine that they are not in balance, in terms of equal areas of responsibility. Similar to the three algorithms in Wang et al. [14], the sensors move to equalize the areas using only local information, such as the distance to neighboring sensor or the areas of neighboring sensor-regions (a sensor's *neighbors* consist of the sensors assigned to each adjacent *sensor-region*). Each sensor seeks to move in a way to reduce the imbalance.

Prior work introduced two algorithms that do this balancing well (Arney et al. [2, 3]). In the first algorithm, called the *largest-neighbor* algorithm, each sensor moves a *fixed distance* toward its neighbor whose sensor-region has the largest area, in an attempt to increase its own area and decrease its neighbor's area. In the second algorithm, called the *weighted-area* algorithm, each sensor moves in a direction found by *weighting the differences in its area and its neighbor areas.* If the sensor has sensor-region area A_i and position $\vec{x}_i$, and its neighbors (comprising the set $N(i)$) have sensor-region areas A_j and positions $\vec{x}_j$, then the entity will adjust its position by

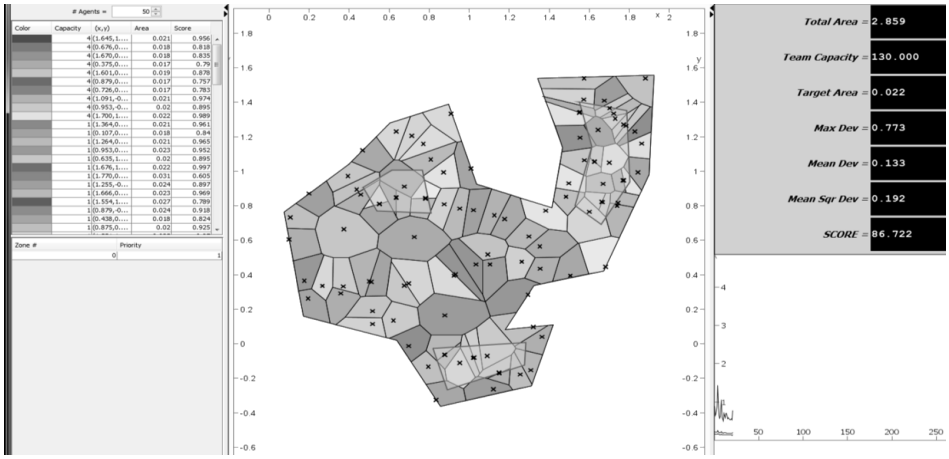
$$\gamma \sum_{j \in N(i)} (A_j - A_i) \frac{\vec{x}_j - \vec{x}_i}{\|\vec{x}_j - \vec{x}_i\|},$$

where γ is a constant nonnegative parameter. This is equivalent to gradient-based local optimization of the function $\sum_{j \in N(i)} (A_j - A_i)(\vec{x}_j - \vec{x}_i)$.

2.4. Equidistribution Platform

A Java application called *Equidistribution Platform* was developed to run the simulations, using the *Blaise* mathematics and visualization Java library (Peterson [12]). The application allows interactive manipulation of the initial polygonal region, the number of sensors, the sensor capabilities (there may be different kinds of sensors), and the geometric priorities for zones of differing importance or sensitivity. Once the initial system is established,

FIGURE 2. Java-based simulation platform for studying geometric distribution algorithms.



the platform generates points at random within the region and displays the corresponding Voronoi partition. The user may select and run an algorithm, watch the movement of sensors, and monitor the metrics of the simulation. An image of the platform interface is shown in Figure 2.

2.5. Algorithm Convergence

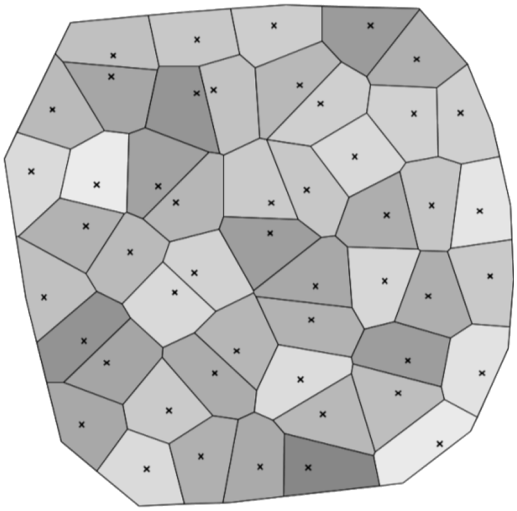
Figure 3 shows a sample converged sensor network (50 sensor-regions of nearly equal area) found by iterating the weighted-area algorithm for the initial geometry and sensor positions in Figure 1. This positioning has a normalized maximum deviation of 0.006, a normalized mean deviation of 0.002, and a normalized mean-squared deviation of 0.002. The utility score is 49.937 out of 50, showing a near-perfect balance in the coverage areas of the 50 sensors. These performance data are typical for the weighted-area algorithm with a simple convex region and homogeneous sensors. Based on experimental evidence, with 100 sensors the normalized mean deviation converges to zero at about order $t^{-0.8}$, where t is the iteration. (Thus, if the initial value of the normalized mean deviation is M, then after t steps it is approximately $Mt^{-0.8}$.) The largest-neighbor algorithm also performs well, but does not exhibit the same kind of robust convergence (Arney et al. [2, 3]).

2.6. Weighted (Prioritized) Regions and Variable Sensor Capabilities

Algorithms and metrics based on sensor-region areas can be generalized to a wider range of scenarios. First, the concept of "region" can be generalized from a homogeneous polygon to a heterogeneous compact region of the plane; the *Equidistribution Platform* allows the user to define one main region R and several "zones" $Z_1, Z_2, \ldots, Z_k$ of varying priorities within the region. The region has a single priority P, and each zone Z_i has priority P_i. The priority of an arbitrary point $\mathbf{x}$ within the region is defined to be $p(\mathbf{x}) = P + \sum_{i \in Z(x)} P_i$, where $Z(\mathbf{x}) = \{i\colon \mathbf{x} \in Z\}$. The *priority-weighted area* of a subset S of the region is $A(S) = \int_{x \in S} p(\mathbf{x})\, dA$. The *priority-weighted area assigned to a sensor* i is $A(S_i)$, where S_i is the sensor-region of i. A key observation in this paper is that a sensor's priority-weighted area works just as well as the sensor-region area in the algorithms and metrics discussed previously. The revised goal of the sensor network is to equidistribute the total priority-weighted area $A(R)$ of the region among all the sensors in the scenario.

Similarly, the algorithms and metrics generalize to variable sensor capabilities. Sensors with greater capacity should pick up a proportionally greater share of the load. Let q_i be a

FIGURE 3. Sensor locations computed using the weighted area algorithm, with 50 sensors arrayed in locations to produce equal areas of responsibility.



capacity parameter defined so that the ability of a sensor i to cover an area is proportional to q_i. Define the *adjusted priority-weighted area assigned to a sensor* i to be $A(S_i)/q_i$.

These two modifications impact the goals of the equidistribution algorithm: now the system seeks to equalize the quantities $A(S_i)/q_i$ for all the sensors. If these values are all equal, then their sum is $A(R)/\sum_i q_i$, the priority-weighted area of the region divided by the total sensor capacity. This is the *target area* for the scenario. Conveniently, the algorithms described previously can also be applied using the target area in place of the mean area, and the adjusted priority-weighted area of a sensor in place of its actual area.

When the simulations converge, sensors in high-priority zones are assigned smaller actual areas (to equalize the weighted areas), and sensors with larger capacities having proportionally larger actual areas. The simulation results described in §3 include scenarios with areas of different priority and sensors with different capabilities. Recall that variable capabilities and priorities do not affect the maximum utility score, which is always equal to the number of sensors in the network. Each sensor contributes a maximum score of 1 when its weighted coverage area is equal to the target area. A sensor's score is reduced by the percent that it misses that target, either above or below the target score.

3. Applications

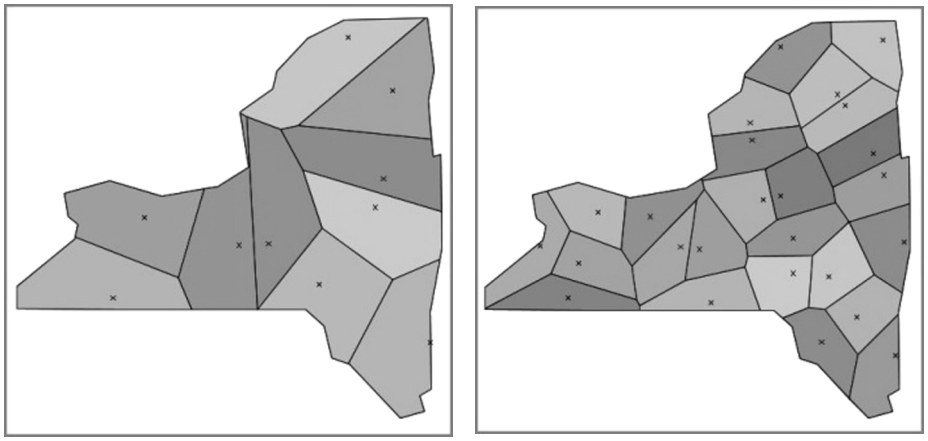
We present six simulations that show the performance and flexibility of the weighted-area algorithm and explain the attributes of the simulation platform.

3.1. Simulation 1—Satellites

One (very) mobile sensor application is the positioning of satellites over the earth so that each covers an equal area. An example of the weighted-area equidistribution algorithm used to distribute 10 and 25 satellites over the coverage region of New York State, starting from random locations, is shown in Figure 4.

The dynamic nature of the algorithm enables the satellites' positions to be autonomously adjusted as satellites are incapacitated or added to the region. For example, programming the satellites with the weighted-area algorithm would allow each satellite to alter its own

FIGURE 4. Result of the converged weighted-area algorithm distributing 10 (left) and 25 (right) satellites over New York State.



Note. Initial distributions are random.

coverage area in response to a change in the operability or coverage of a neighbor. This would be done autonomously by the satellites and would not require an outside central controller to recalculate the necessary coverage and then program each satellite to move to a new designated location.

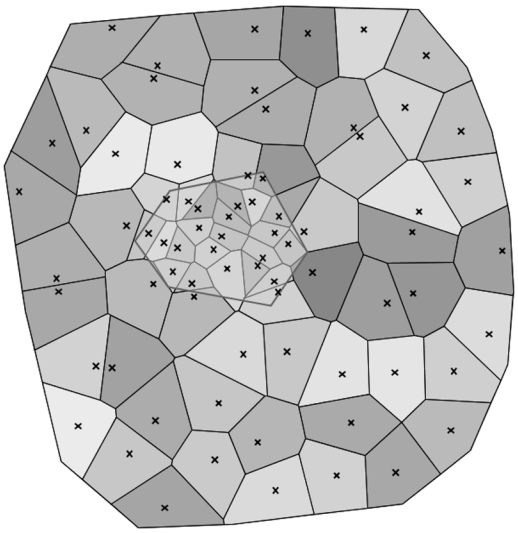
3.2. Simulation 2—Zones of Higher Priority

In this simulation, we embed a zone that is five times more important than the overall coverage region. This kind of prioritization could occur in various scenarios in which intense surveillance or monitoring of a zone is required due to highly important or highly sensitive information, equipment, or facilities. This scenario calls for 75 sensors to be placed in the entire region, and their locations provide balanced coverage of both the large overall region and five times more dense sensor coverage in the higher-priority zones. Figure 5 shows the result of iterating the weighted-area algorithm 100 times. The density of sensors in the higher-priority zones is approximately five times that of the outer region. The assigned coverage areas of the inner-zone sensors are therefore 1/5 of the area of the outer region sensors. The performance metrics indicate a mean deviation of 0.007 and a utility score of 74.4 out of 75. This simulation shows that the algorithm is able to generate effective sensor coverage in a scenario with a high-priority zone.

3.3. Simulation 3—Sensors with Different Capabilities

Sensors with greater capacity should be given more responsibility in terms of coverage area. In this simulation of covering a region with 75 mobile sensors, we give 14 sensors six times the capacity of the other 61 sensors. Therefore, the total sensor network capacity is $145 = 14 \times 6 + 61 \times 1$. In the resulting coverage map shown in Figure 6 (found by iterating the weighted-area algorithm starting from random sensor locations), the 14 high-capacity sensors have larger areas of responsibility. In this case, the performance metrics of the algorithm do not converge as quickly. The mean deviation is 0.319 and the utility score is approximately 52 out of 75. This lower performance is due to the extreme variation in sensor capacities; the use of the Voronoi partition is not completely compatible with this large variation, since it assumes that the coverage capability of a sensor depends only on the distance from the sensor.

FIGURE 5. Sensor locations computed using the weighted area algorithm, with 75 sensors arrayed in locations to produce equal weighted areas of responsibility based on the five times higher priority for the outlined inner zone.



3.4. Simulation 4—Higher Priority and Sensors with Different Capabilities

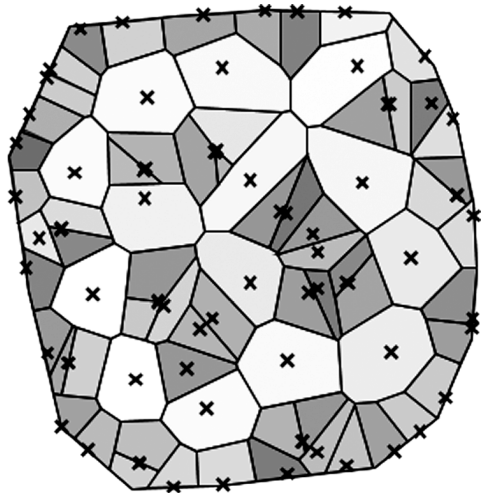
The fourth simulation combines both a higher priority region (from Simulation 2) and higher capable sensors (from Simulation 3). We have 75 sensors (10 at six times the capability as the others) and one zone with five times the importance of the outer region. The results show a mean deviation of 0.279 and a utility score of 54.095 out of 75. We show the sensor deployment in Figure 7 with the local links (sensor to its neighbors) of the sensor network shown on the figure.

3.5. Simulation 5—Dynamics in the Battlespace

During sensor network operations many changes could affect the performance and coverage of the sensor network. Sensors could fail, new sensors with different capabilities could be employed, a high priority zone could revert to lower priority (or vice versa), and new priorities could develop. In this simulation, we see all those things happen to the sensor network. In particular, our simulation begins with 100 randomly positioned sensors (90 with standard capacity c and 10 with four times the capacity $4c$) that need to cover a nonconvex region that contains three zones with varied and higher priorities. See Figure 8 (left) for the converged sensor array covering this situation. After iterating the weighted-area algorithm, the performance metrics show a mean deviation of 0.137 and a utility score of 86.4 out of 100.

In the next phase of the operation, the situation and the need for sensors change: (1) 30 sensors are added, four of which have a capacity four times the standard capacity (there are now 116 sensors with standard capacity c and 14 with four times the capacity $4c$); (2) a fourth higher-priority zone is added; (3) one previous zone changes shape; and (4) another original zone changes priority. The new sensor coverage array is shown in the right side of Figure 8. None of these changes pose difficulty for this adaptive cooperative algorithm. The performance metrics however indicate a mean deviation of 0.257 and a utility score of 96.6 out of 130. The degraded metrics for both simulations shown in Figure 8 are because the region is nonconvex and there are numerous sensors with varied capabilities.

FIGURE 6. Sensor locations computed using the weighted area algorithm, with 75 sensors arrayed in locations to produce equal weighted areas of responsibility based on the 14 sensors with six times the capacity of the standard sensors.



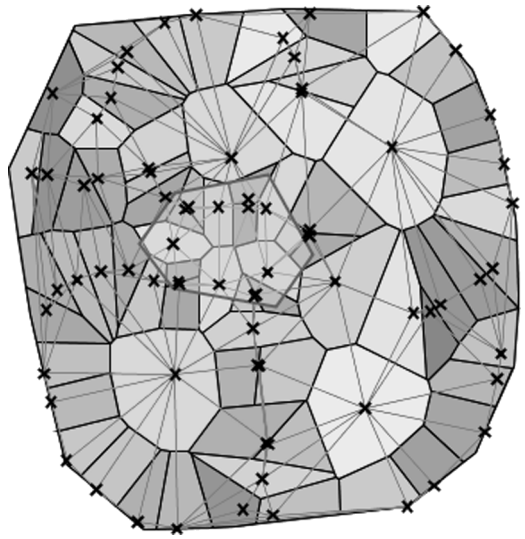
Note. The 14 high-capacity sensors are easy to locate since their regions of responsibility are visibly much larger than the standard sensors.

The *Equidistribution Platform* allows the user to specify arbitrary regions and priority zones. Figure 9 shows the GUI for entering zones and priorities, as used in the setup of the simulation in Figure 8.

3.6. Simulation 6—Practical Geometries, Priorities, and Capabilities

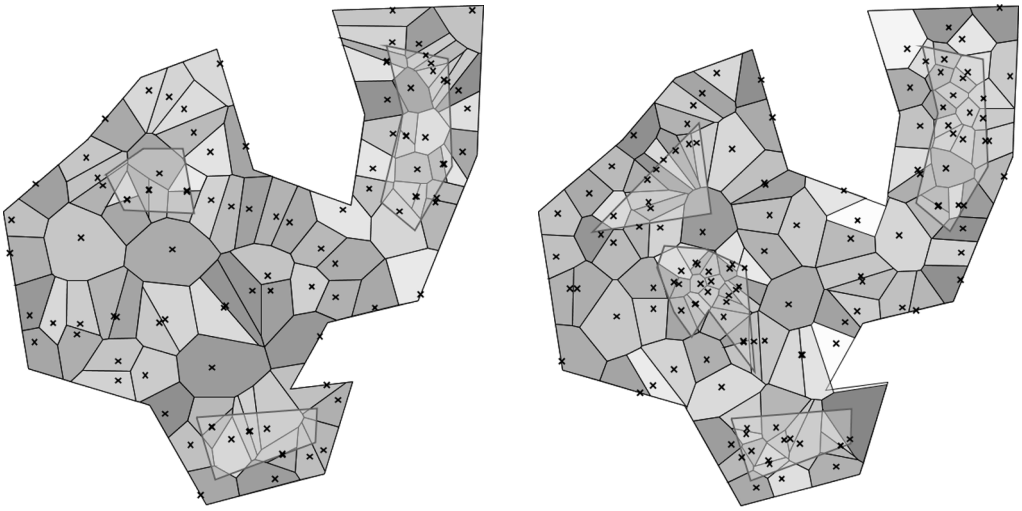
Real operations often involve more complexity than simple simulations allow. This final simulation attempts to include some of the complexities of an operational situation. The

FIGURE 7. Deployment of 75 sensors with 10 sensors with six times the capacity of the standard sensors and an outlined zone of five times the importance.



Note. The resulting sensor-neighbor network is shown on the deployment map.

FIGURE 8. The left diagram shows the converged sensor array with 100 sensors (10 with higher capability) with three higher-priority zones; after the situation changes, the right part of the figure contains a sensor network of 130 sensors (14 with the four times higher capability) covering four higher-priority zones, including one that has changed shape, another that has a changed priority, and an entirely new high-priority zone.



outer region in this simulation is a map of the countries of Afghanistan, Tajikistan, and Kyrgyzstan. With the intent to show how sensors could be deployed to monitor operational situations such as traffic along the Northern Distribution Network (NDN) from Transit Center Manas in Kyrgyzstan, through Tajikistan, and southward to the supply routes in Afghanistan. That NDN zone is given a priority of six times the overall region and is outlined in the left half of Figure 10. In that figure, we show the algorithm's coverage of the three countries and the NDN zone by 150 sensors. The mean deviation in the balance of coverage is 0.42 and the utility score is 87 out of a possible 150. We then add three more priority zones and 50 more standard sensors to the network. Two of the new zones are border

FIGURE 9. The interface of the *Equidistribution Platform* used to specify the region, zones, and priorities.

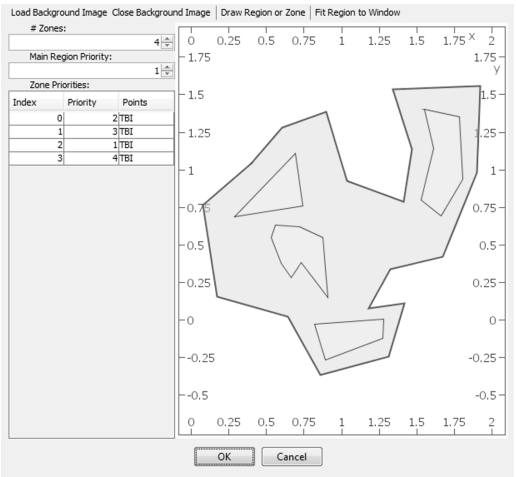


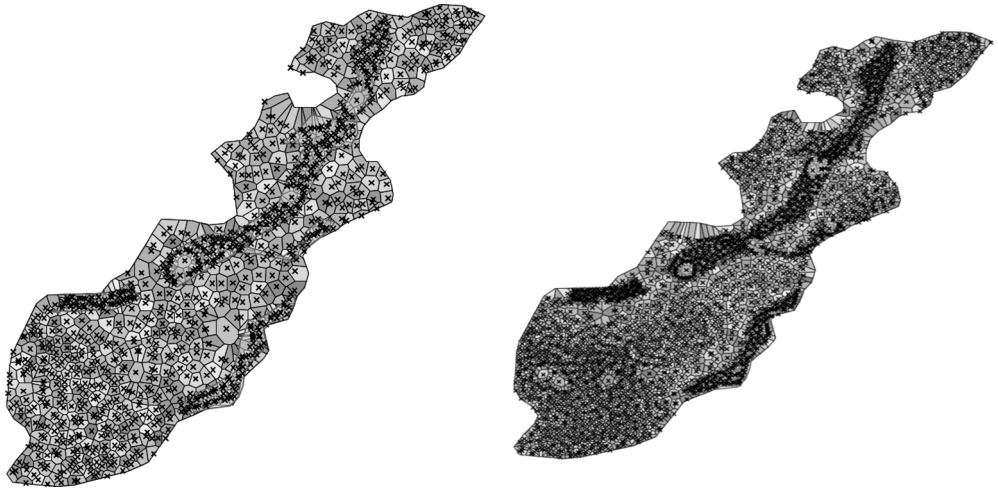
FIGURE 10. The left diagram shows 150 sensor locations and area coverage of the three countries with a higher-priority NDN zone; the right shows 200 sensors covering an additional three priority regions (border areas between the three countries with priority of four and the subregion along the Afghanistan-Pakistan border with priority of seven).



Notes. The NDN zone is given a priority of six times the overall region and is outlined in the left diagram.

areas between the three countries given priority of four and the other zone is along the Afghanistan-Pakistan border and is assigned a priority value of seven. This new sensor array is shown in the right part of Figure 10. The mean deviation of the sensor areas for this new network is 0.397 and the utility score is 120.7 out of 200. These sensors could be static implanted sensors or mobile platforms and the sensors' modes could be any mode, such as radar, seismic, acoustical, optical, or chemical. The sensors could be employed in different roles (early warning, detection, tracking, identification) and different modes as long as their capabilities could be ranked and used in the deployment algorithm.

FIGURE 11. The left diagram shows sensor-locations for coverage of the three Central Asia countries with five higher-priority zones by 1,000 sensors; the figure on the right shows 2,500 sensor-locations covering these same areas.



To demonstrate more realism and dynamics, we add another high priority area in this simulation along the Afghanistan-Turkmenistan border at priority six and increase the number of sensors to 1,000 (shown on the left) and 2,500 (shown on the right) of Figure 11. We also add 11 higher capability sensors, which are noticeable in the results of both simulations for their larger coverage areas. For the 1,000 sensor simulation, the mean deviation is 0.359 and the utility score is 641 out of 1,000. For the 2,500 sensor simulation, the mean deviation is 0.375 and the utility score is 1,563 out of 2,500. This simulation shows both the utility and flexibility of the weighted-area algorithm with large numbers of sensors for efficient and effective coverage of real geographical areas.

4. Conclusions

The equidistribution simulation results indicate that the locally autonomous weighted-area algorithm is sufficient to achieve a good global balance in coverage areas assigned to the deployment of heterogeneous sensors. We have demonstrated that cooperative algorithms based upon this kind of geometric structure can be used for sensor deployment in dynamic situations to include the assignment of additional sensors with different capabilities, the inoperability of a sensor, new priorities and geometries in the coverage, and changes in the area of coverage. The algorithm is highly distributed and requires little in the way of computation or communication from any individual sensor. The balance of coverage, flexibility, efficiency, and effectiveness of our deployment algorithms are encouraging and could lead to better performance in many aspects of sensor networks.

Acknowledgments

The authors would like to thank the referee for many valuable comments and suggestions. This work was completed while the third author was under the support of an NRC/Davies postdoctoral research grant and an ARO research grant.

References

[1] D. C. Arney and E. Peterson. Cooperation in social networks: Communication, trust, and selflessness. host organization. *Proceedings of the 26th Army Science Conference, Orlando, Florida*, 2008. http://handle.dtic.mil/100.2/ADA505995. Accessed February 7, 2011.

[2] D. C. Arney, K. Arney, and E. Peterson. Models and metrics of geometric cooperation. *Proceedings of the Winter Simulation Conference, Baltimore, MD*, IEEE, 1383–1393, 2010.

[3] D. C. Arney, K. Arney, and E. Peterson. Autonomous geometric cooperation: Pursuit-Evasion, equi-distribution, surveillance. *Proceedings of the 27th Army Science Conference, Orlando, FL*, http://www.armyscienceconference.com/manuscripts/E/EO-002.pdf. Accessed February 7, 2011.

[4] F. Aurenhammer. Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Computing Surveys* 23:345–405, 1991.

[5] J. Ben, X. Tong, Y. Zhang, and H. Zhang. Discrete global grid systems: Generating algorithm and software model. *Geoinformatics* 6421:1–13, 2006.

[6] Z. J. Butler, A. A. Rizzi, and R. L. Hollis. Contact sensor-based coverage of rectilinear environments. *IEEE International Symposium on Intelligent Control/Intelligent Systems and Semiotics, Cambridge, MA*, IEEE, 266–271, 1999.

[7] C. Frank and K. Römer. Distributed facility location algorithms for flexible configuration of wireless sensor networks. *Distributed Computing in Sensor Systems. Proceedings of the 3rd IEEE International Conference,* Springer-Verlag, Berlin, 124–141, 2007.

[8] C. W. Kang and M. W. Golay. An integrated method for comprehensive sensor network development in complex power plant systems. *Reliability Engineering and System Safety* 67(January): 17–27, 2000.

[9] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. *Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, Anchorage, AK*, IEEE, 1380–1387, 2001.

[10] R. Mulligan and H. M. Ammari. Coverage in wireless sensor networks: A Survey. *Network Protocols and Algorithms* 2:27–53, 2010.

[11] T. Ohyama. Division of a region into equal areas using additively weighted power diagrams. *IEEE Proceeding of the 4th International Symposium of Voronoi Diagrams in Science and Engineering, Glamorgan, UK*, IEEE, 2007.

[12] E. Peterson. Blaise mathematics and visualization library. http://blaisemath.googlecode.com. Accessed December 18, 2010.

[13] J. Snyder. An equal-area map projection for polyhedral globes. *Cartographica* 29:10–21, 1992.

[14] G. Wang, G. Cao, and T. LaPorta. Movement-assisted sensor deployment. *Twenty-third Conference of the IEEE Computing Society on Mobile Comuting, Hong Kong*, 5:620–652, 2004.