

Research Article

Cluster Density of Dependent Thinning Distributed Clustering Class of Algorithms in Ad Hoc Deployed Wireless Networks

Sankalpa Gamwarige and Chulantha Kulasekera

Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa 10400, Sri Lanka

Correspondence should be addressed to Sankalpa Gamwarige, sgamwarige@gmail.com

Received 21 October 2011; Accepted 13 February 2012

Academic Editor: Youyun Xu

Copyright © 2012 S. Gamwarige and C. Kulasekera. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Distributed clustering is widely used in ad hoc deployed wireless networks. Distributed clustering algorithms like DMAC, HEED, MEDIC, ANTCLUST-based, and EDCR produce well-distributed Cluster Heads (CHs) using dependent thinning techniques where a node's decision to be a CH depends on the decision of its neighbors. An analytical technique to determine the cluster density of this class of algorithms is proposed. This information is required to set the algorithm parameters before a wireless network is deployed. Simulation results are presented in order to verify the analytical findings.

1. Introduction

Distributed clustering is a robust technique used to organize ad hoc deployed wireless nodes to form a communication network [1]. It is widely adopted in energy constrained ad hoc deployed wireless sensor networks (WSNs) [2]. Distributed clustering algorithms, used in ad hoc deployed wireless networks, can be broadly categorized into two classes. The first category consists of independent randomized cluster head (CH) selection class of algorithms; that is, the decision for a node to be a CH is made independent of the decision of its neighboring nodes. For example algorithms such as LEACH [3], LEACH-D [4], SEP [5], and EDAC [6] fall into this category. These algorithms do not produce well-distributed CHs [7]. They may produce two or more adjoining nodes as CHs. Furthermore, the variation between the theoretical expected number of CHs for these algorithms is considerable when compared to the actual number of CHs obtained after deployment [8]. The second category consists of distributed clustering algorithms like DMAC [1], HEED [9], ANTCLUST based [10], MEDIC [11], EDCR [12], and its derivatives [13]. The location of a CH for these algorithms is dependent on its neighbors decision as well. This ensures that no two CHs appear in each others neighborhood and all nodes have at least one CH in their neighborhood or the node itself is a CH. They produce well-distributed clusters using dependent decision making and is referred too as

Dependent Thinning Distributed Clustering (DTDC) class of algorithms. We note that the CH selection process of DTDC class of algorithms resembles the *reverse price auction* and is sometimes known as the Dutch auction [11] method.

Irrespective of the distributed clustering algorithm used in ad hoc deployed wireless network applications, the knowledge of the expected number of clusters, denoted by $E[k]$, is an important parameter required at the planning stage of the network. For example, consider a WSN where data is collected periodically and aggregated at the CH, then communicated to the base station (BS). The application may expect $E[k]$ number of clusters, where each cluster has an expected number of nodes, denoted by $E[n]$, in the given deployment area A . The given requirement is generated based on the level of reliability expected from the collected data. That is, the reliability is directly connected to the redundancies associated with the nodes within a cluster [14]. Another example is if an ad hoc deployed wireless network application is required to produce an optimal number of clusters $E[k]$ based on the requirement to minimize the energy cost for communication and maximize the network lifetime [15, 16]. In both these examples, the WSN parameters should be set appropriately at the initial deployment stage so that when in operation the desired number of clusters $E[k]$ is achieved to meet the design objective. In the first example, this objective is increased

reliability, where as in the second example it is to maximize the lifetime.

To identify the importance of knowing WSN design parameters, let us look at a known example like the LEACH algorithm from the first category. LEACH uses a parameter p which represents the expected proportion of nodes to be CHs. That is, a node has p probability of becoming a CH independent of the decision of its neighbors. When an independent randomized clustering algorithm like LEACH is applied to an ad hoc deployed network where N nodes are uniform randomly deployed in a given area, the expected number of clusters can be found using the expression $E[k] = pN$ [3]. According to [17], the node distribution of such a system is considered to be a 2-D Poisson point process with intensity $\lambda = N/A$, and the resultant CHs too would be distributed as a 2-D Poisson point process with intensity (i.e., CH density) $\lambda_c = p\lambda$. We see that by setting the WSN parameters p and N we can achieve a desired $E[k]$. That is the analytical expressions presented play an important part in achieving the proper $E[k]$.

To the authors knowledge, no such analysis exists to determine the CH distribution and density (λ_c) of DTDC class of algorithms. However, Bettstetter [18] has presented an empirical formula for the CH density of the DMAC algorithm using simulation results. As it is an empirical formula, it cannot be generalized. In this paper we present an analytical expression for CH density for the DTDC class of algorithms in order to address this gap.

In what follows, we will first establish that the DTDC class of algorithms such as HEED, ANTCLUST based, DMAC, MEDIC, and EDCR will indeed fall into one common category in terms of their CH distribution. Then, we will determine the probability distribution of the cluster area of the DTDC class of algorithms. Subsequently, the distribution of the cluster area will be used to derive the cluster density. Furthermore, we will also consider the boundary (or boarder) effect due to the finite geographical area in which the nodes are distributed and modify the expressions to accommodate it. The proposed analytical results will prove that the empirical results derived using simulations by Bettstetter in [18] are indeed accurate.

Rest of the paper is organized as follows: Section 2 presents the nomenclature. Section 3 provides a mathematical model to express the CH selection and distribution common to all DTDC algorithms. In in Section 4 the model presented in the previous section will be used to identify a probability distribution of cluster area of DTDC class of algorithms. Subsequently in Section 5 these results will be used to find the cluster density and the number of expected clusters in a rectangular and circular deployment area. Simulation results presented in Section 6 establish that the analytical findings are in line with the actual values presented in existing literature. Section 7 presents the conclusion.

2. Nomenclature

Table 1 gives the notations used in what follows. Some are extracted from [12].

TABLE 1: Brief definition of some notations used in the paper.

λ	Intensity of sensor node distribution point process.
α	Random variable denoting the cluster area.
R	CH message broadcasting distance (using an omni direction antenna).
T_i	Time at which a node i is supposed to broadcast its CH candidacy, provided that it has not heard similar messages from its neighbor.
\mathcal{S}	Set of all N sensor nodes deployed in the sensor bed.
\mathcal{H}	Set of all CHs at a given moment.
$ n_i - n_j $	Euclidean distance between any two nodes n_i and n_j .
y	Random variable denoting the probability of any selected random node could be a CH in a given moment.
$E[k]$	Expected number of clusters where k represents the number of clusters in any given moment.
$E[n]$	Expected number of nodes in a given cluster.
λ_c	CH density.

3. Preliminaries

This section presents the background necessary to find the CH density of DTDC class algorithms covering HEED, DMAC, ANTCLUST base, MEDIC, and EDCR. As mentioned before, these algorithms will produce well-distributed CHs by making a node's decision to be a CH based on the decisions of other nodes in its neighborhood.

We assume that there are N number of uniform-randomly distributed nodes in a given deployment area A resulting in a 2-D Poisson point distribution of intensity λ , where $\lambda = N/A$ [17] in our analysis. Furthermore we assume that all clusters are well populated; that is, each cluster consists of a large number of less reliable low cost nodes which work collaboratively to achieve reliable results. Hence, $\alpha\lambda \gg 1$ where α is a random variable denoting the cluster area. According to the DTDC class of algorithms, the area covered by a CH candidacy message is given by πR^2 , where R represents the maximum distance a CH candidacy announcement message would reach. Since $\alpha < \pi R^2$,

$$\pi R^2 \lambda \gg 1. \quad (1)$$

The following common features exists in the DTDC class of algorithms.

- (a) DTDC class of algorithms does not allow two CHs to be within a distance R . Furthermore, it ensures that all the nodes are either discovered by a CH (i.e., there is a CH within a distance R of a regular node), or the node itself is a CH.
- (b) Each node calculates a time T_i at which it will broadcast the CH candidacy announcement, provided it has not heard a similar message from a neighbor by this time. T_i calculation is algorithm specific. However, all algorithms ensure that T_i is inversely proportional to the fitness of a node to be a CH. For example in the EDCR algorithm, T_i is inversely

proportional to the relative residual energy level of a node [12]. As such, the node with the highest fitness to become a CH will have the lowest T_i , resulting it to announce CH candidacy first and becoming the CH for that neighborhood.

- (c) All the algorithms use a random component for tiebreaking. Hence, when all nodes are equally fit to be CHs, T_i is purely random. This is true for EDCR, HEED, MEDIC and ANTCLUST algorithms at the initial deployment stage since all nodes have equal energy.

Above features of DTDC class of algorithms reaffirm that the selected CHs represent a dependent thinning point process on the original 2-D Poisson point process. Let \mathcal{S} represents the set of all deployed nodes, where $\mathcal{S} \subset \mathbb{R}^2$ with $|\mathcal{S}| = N$. The clustering process yields a random set $\mathcal{H} \subseteq \mathcal{S}$ of secondary points which are CHs with the property that $|h_i - h_j| > R$, where $h_i, h_j \in \mathcal{H}$ and $i \neq j$. Note that $\mathcal{S} \setminus \mathcal{H}$ are the regular (non-CH) member nodes. For any node $m_k \in \mathcal{S} \setminus \mathcal{H}$ we have $|m_k - h_i| < R$ and $T_{m_k} > T_{h_i}$ at least for one CH node $h_i \in \mathcal{H}$. Further, it should be noted that m_k is a member of the cluster with CH h_i when $|m_k - h_i| < |m_k - h_j| < R$ for all $i \neq j$.

According to [19] aforementioned dependent thinning point process follows a Matérn Type III process when T_i is a pure random value. Hence, we can conclude that the CH distribution of dependent thinning algorithms like HEED, ANTCLUST, DMAC, MEDIC, and EDCR immediately after deployment would resemble a Matérn Type III point process.

Example 1. Figure 1 gives a simplified description of Matérn Type III process applied to 3 random nodes a , b , and c with $|a - b| = 0.30$, $|b - c| = 0.23$, $|a - c| = 0.53$, $R = 0.4$, $T_a = 0.27$, $T_b = 0.52$, and $T_c = 0.78$.

According to this illustration, since $T_a < T_b$, a eliminates b ; since b is eliminated, even though $T_b < T_c$, c would not be eliminated; hence, nodes a and c will be elected as CHs. Even though the above description clearly indicates that the DTDC class of algorithms resemble a Matérn Type III process, we cannot find the resultant CH density (or expected number of clusters) using this information. As Bertil Matérn has shown in [20], the point distribution of Matérn Type III-dependent thinning process is mathematically intractable.

Based on this background, we will derive the CH density of the class of DTDC algorithms by finding the probability density function (p.d.f) of α for practical cases satisfying (1) in the next section.

4. Probability Density Function of Cluster Area

Based on our analysis we observe that the probability of α depends on the following two scenarios.

- (1) For a given cluster area, there are no uncovered nodes (uncovered node means a node that has not heard from a neighboring CH almost at the end of a new CH candidacy announcement time interval) in its cluster neighborhood.

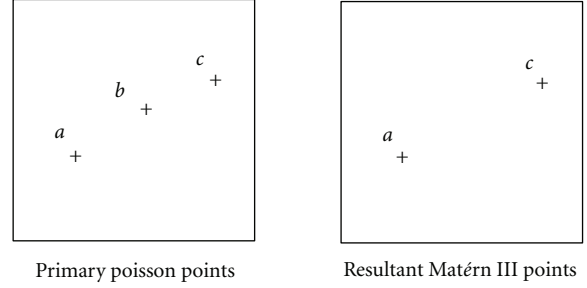


FIGURE 1: Simple example of Matérn Type III repulsive point process.

- (2) The chance of having no such uncovered nodes.

Let P_B be the probability that no uncovered nodes exist in a given cluster neighborhood. Then the conditional probability $P_{A|B}$ denotes the cluster area given no uncovered nodes exist in a given cluster neighborhood. Based on these facts, we find the probability $P_{A \cdot B}$ of a resultant cluster area α when no uncovered nodes exists. One finds that

$$P_{A \cdot B}(\delta_1 \leq \alpha \leq \delta_2) = P_{A|B}(\delta_1 \leq \alpha \leq \delta_2) \cdot P_B(\delta_1 \leq \alpha \leq \delta_2), \quad (2)$$

where $0 \leq \delta_1 < \delta_2 \leq \pi R^2$.

We use Figures 2, 3, and 4 to explain (2). Please note that the radius of each disk is R in all the figures.

According to the class of DTDC algorithms, smallest possible cluster area size would result whenever a given CH's neighboring CHs sit on the perimeter of its CH broadcasting coverage disc of radius R since no two CHs could be selected within each other's CH broadcasting range R . This situation is shown in Figure 2.

Hence, we can write

$$P_{A|B}\left(0 < \alpha < \frac{\sqrt{3}R^2}{2}\right) = 0. \quad (3)$$

In other words, Figure 2 shows the possible highest CH density (Number of CHs in a given unit area). According to the DTDC class of algorithms, we can expect cluster area sizes between smallest of $\sqrt{3}R^2/2$ to largest of πR^2 provided that there are no uncovered nodes in the cluster neighborhood. Therefore, we can write

$$0 < P_{A|B}(\delta_3 \leq \alpha \leq \delta_4) \leq 1, \quad (4)$$

where $\sqrt{3}R^2/2 \leq \delta_3 < \delta_4 \leq \pi R^2$.

Further, when we have close packed clusters (smallest as shown in Figure 2 and largest as shown in Figure 3), there cannot be any uncovered areas. In other words, when cluster area $\alpha > 3\sqrt{3}R^2/2$, there can be uncovered nodes in its neighborhood since there can be uncovered neighboring regions as shown in Figure 4.

$P_B(\alpha)$ represents the probability that there is no uncovered nodes in a given cluster (with area α) neighborhood. This can be expressed by

$$P_B(\alpha) = P(n = 0 | \lambda A_u) = e^{-\lambda A_u}, \quad (5)$$

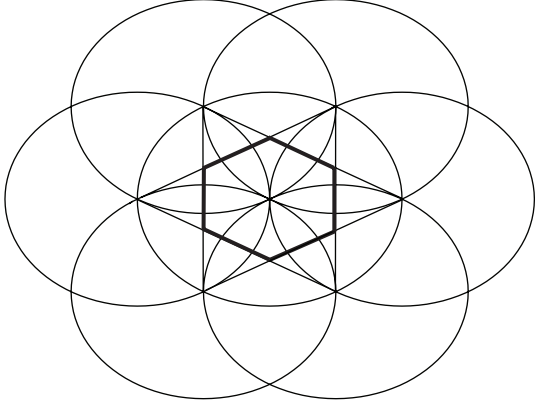


FIGURE 2: Smallest possible cluster size is $\sqrt{3}R^2/2$.

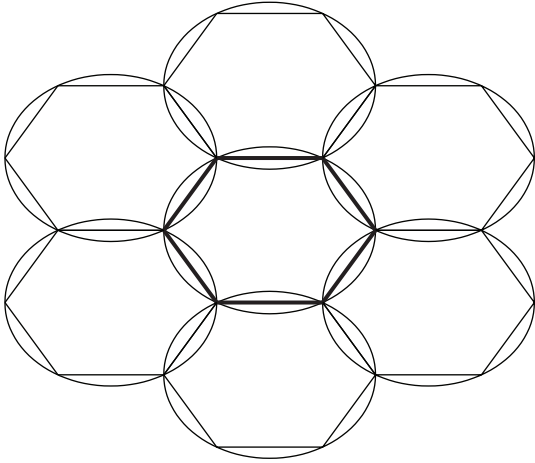


FIGURE 3: Largest possible closed packed cluster size is $3\sqrt{3}R^2/2$.

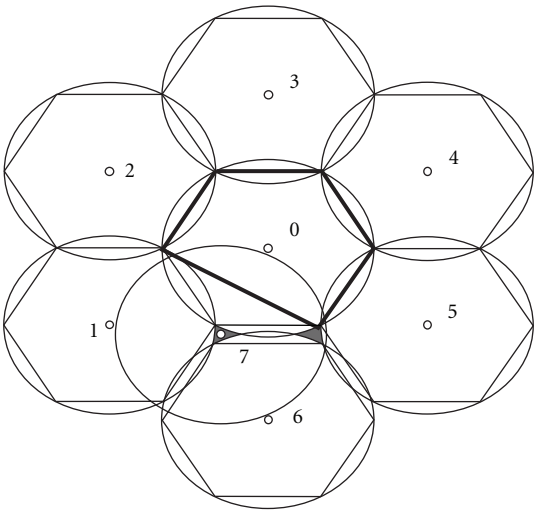


FIGURE 4: Cluster area more than $3\sqrt{3}R^2/2$ creates uncovered region with area A_u (shaded in gray).

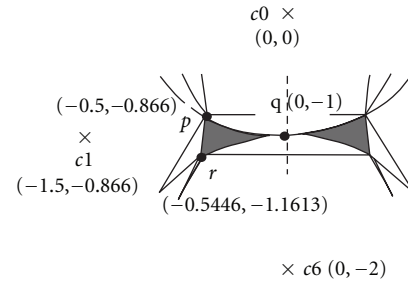
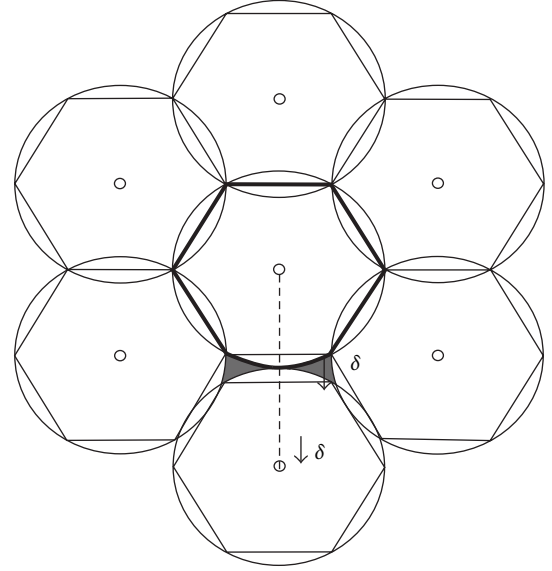


FIGURE 5: Proof of $P_B(\alpha > 3\sqrt{3}R^2/2) \rightarrow 0$.

where A_u is any uncovered area formed by the cluster setup as shown in Figure 4. We can show that the neighboring clusters are close packed when the cluster area, $\alpha \leq 3\sqrt{3}R^2/2$. In other words, there is no uncovered area, resulting in $A_u = 0$ for $\alpha \leq 3\sqrt{3}R^2/2$. As a result, the probability that there would not be any uncovered nodes is given by

$$P_B\left(\alpha \leq \frac{3\sqrt{3}R^2}{2}\right) = 1. \quad (6)$$

According to (5), $P_B(\alpha)$ is an exponential decaying function when $\alpha > 3\sqrt{3}R^2/2$. Now let us consider Figure 5. This is a special case of Figures 3 and 4 where nodes 0 and 6 are placed $2R$ distance apart. According to Figure 5, there is a chance for a node to be in the uncovered area A_u shaded in gray. The cluster area α of Figure 5 can be expressed as

$$\alpha = \frac{3\sqrt{3}R^2}{2} + \left(\frac{\pi}{6} - \frac{\sqrt{3}}{4}\right)R^2 = \frac{3\sqrt{3}R^2}{2}(1 + .0349). \quad (7)$$

This is only 3.49% bigger than the size of the cluster area shown in Figure 3. The uncovered area A_u of Figure 5 is

$$A_u = 2\left(\Delta_{p,q,r} + \Delta_{p,q,c0} + \Delta_{p,r,c1} + \Delta_{q,r,c6} - S_{p,q,c0} - S_{p,r,c1} - S_{q,r,c6}\right), \quad (8)$$

where, in general, $\Delta_{x,y,z}$ represents an area of a triangle $\{x, y, z\}$, and $S_{x,y,z}$ represents an area of a sector $\{x, y, z\}$. Since $p = (-0.5R, -0.8660R)$, $q = (0, -R)$, $r = (-0.5446R, -1.1613R)$, $c0 = (0, 0)$, $c1 = (-1.5R, -0.8660R)$ and $c6 = (0, -2R)$, we can derive $A_u = 0.094R^2$.

We have shown that $\lambda\pi R^2 \gg 1$ in (1). Therefore, if we consider a WSN with 100 nodes in a given node neighborhood, then $\lambda\pi R^2 = 100$ and the resultant $P_B = P(n = 0 \mid \lambda A_u) = 0.0502$. On the other hand, when the neighborhood contains 200 nodes, this will be further reduced to $P_B = P(n = 0 \mid \lambda A_u) = 0.0025$. Hence, we can conclude that

$$P_B\left(\alpha > \frac{3\sqrt{3}R^2}{2}\right) \rightarrow 0, \quad (9)$$

where $\lambda\pi R^2 \gg 1$.

Therefore, we can approximate that

$$\begin{aligned} P_B\left(\alpha \leq \frac{3\sqrt{3}R^2}{2}\right) &= 1, \\ P_B\left(\alpha > \frac{3\sqrt{3}R^2}{2}\right) &= 0 \end{aligned} \quad (10)$$

provided that $\lambda\pi R^2 \gg 1$. Hence, once we combine (2), (3), (4), and (10) we obtain that

$$P_{A \cdot B}\left(\frac{\sqrt{3}R^2}{2} > \alpha > \frac{3\sqrt{3}R^2}{2}\right) = 0. \quad (11)$$

Therefore,

$$P_{A \cdot B}\left(\frac{\sqrt{3}R^2}{2} \leq \alpha \leq \frac{3\sqrt{3}R^2}{2}\right) = 1. \quad (12)$$

The resultant cluster areas α of DTDC class of algorithms have an equal chance to be in the interval $[\sqrt{3}R^2/2, 3\sqrt{3}R^2/2]$, due to the fact that all nodes having an equal chance to get the lowest T_i as they may have equal fitness to be a CH. This results in cluster area p.d.f, $p_{A \cdot B}(\alpha)$ to be uniform. Hence,

$$p_{A \cdot B}(\alpha) = \begin{cases} \frac{1}{\sqrt{3}R^2}, & \frac{\sqrt{3}R^2}{2} \leq \alpha \leq \frac{3\sqrt{3}R^2}{2} \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

provided that $\pi R^2 \lambda \gg 1$.

This far we have derived the p.d.f of cluster area α . This result will be used in deriving the expected cluster density in the subsequent section.

5. Derivation of Expected Cluster Density

In this section, we will derive the expected cluster density (or CH density as each cluster is served by one and only one CH) for the class of DTDC algorithms.

Let us define y as the probability that a randomly chosen node is a CH. Thus,

$$\begin{aligned} y &= \frac{\text{Number of CHs in a given area}}{\text{Number of all nodes in the same given area}} \\ &= \frac{1}{\text{Number of nodes in a random cluster}} \\ &= \frac{1}{\alpha\lambda}. \end{aligned} \quad (14)$$

We note that when $z = f(x)$ and x is a random variable with a p.d.f of $p_X(x)$, then the p.d.f of z is given by

$$p_Z(z) = \frac{p_X(f^{-1}(z))}{|f'(f^{-1}(z))|}. \quad (15)$$

We can write the p.d.f of random variable y , $p_Y(y)$ using (13) as,

$$p_Y(y) = \begin{cases} \frac{1}{\sqrt{3}y^2\lambda R^2}, & \frac{2}{3\sqrt{3}\lambda R^2} \leq y \leq \frac{2}{\sqrt{3}\lambda R^2} \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

According to (14), $y = k/N$, where k is the total number of CHs at a given moment, and N is the total number of nodes. Hence $E[y]$, the expected probability that a given node is a CH, can be given as

$$E[y] = E\left[\frac{k}{N}\right] = \frac{E[k]}{N} = \frac{\lambda_c}{\lambda}, \quad (17)$$

where λ_c is the CH density. So we have

$$\lambda_c = E[y]\lambda, \quad (18)$$

$$\begin{aligned} E[y] &= \int_{-\infty}^{\infty} y p_Y(y) dy \\ &= \int_{2/3\sqrt{3}\lambda R^2}^{2/\sqrt{3}\lambda R^2} y \frac{1}{\sqrt{3}y^2\lambda R^2} dy \\ &= \frac{1}{(\sqrt{3}/\pi \ln 3)(\pi\lambda R^2)} \\ &= \frac{1}{0.5018 (\pi\lambda R^2)}. \end{aligned} \quad (19)$$

According to (19), we can expect a 0.5018 fraction of nodes belonging to a given CH's broadcasting range R neighborhood to join its cluster.

Further, using (18) and (19), we can show that

$$\lambda_c = \frac{\lambda}{(\sqrt{3}/\pi \ln 3)\pi\lambda R^2} = \frac{\ln 3}{\sqrt{3}R^2} = \frac{1}{0.5018\pi R^2}. \quad (20)$$

Hence, we can conclude that the expected CH density λ_c is independent of the node density provided that $\pi R^2 \lambda \gg 1$.

Observation 1. The result obtained in (20) matches with the empirical formula proposed by Bettstetter in [18] where $\lambda_c = \lambda/(1 + \mu/2)$ and $\mu = \pi R^2 \lambda$. When $\pi R^2 \lambda \gg 1$ the empirical formula proposed by Bettstetter reduces to

$$\lambda_c = \frac{\lambda}{0.5\mu} = \frac{1}{0.5\pi R^2}. \quad (21)$$

In the analysis thus far we have ignored the influence of the node deployment region boundary and its effects. In what follows, we will analyze the *boundary effect*. The CHs closest to the boundary does not have any neighboring CHs beyond the boundary; that is, nodes at the boundary have a higher isolation probability even though all the nodes are uniformly distributed within the deployed area. Hence, CHs are more likely to be found at the boundary. This was observed and confirmed in [18].

We can use (17) and (19) to derive the expected number of clusters $E[k]$ to be formed assuming that the boundary effect does not exist. In other words, we have relaxed the reality that there can be more CHs close to the boundary compared to rest of the area. Thus,

$$E[k] = NE[y] = \frac{N}{(\sqrt{3}/\pi \ln 3)M}, \quad (22)$$

where $M = \pi R^2 \lambda$ is the expected number of nodes in any given CHs broadcasting range R . That is, in (22), we have not considered the boundary effect. In what follows, we will derive M considering the boundary effect for frequently considered node deployment region shapes, namely, a rectangular region and a circular region. Subsequently we will use these results to obtain $E[k]$ accounting for the boundary effect.

5.1. Boundary Effect on $E[k]$ due to a Rectangular Deployment Area. We derive M for a rectangular region with dimensions $a \times b$ and N ad hoc deployed nodes. For this scenario, the probability (P_0) that two uniformly distributed nodes each within CH candidacy broadcasting range R is given by the integral

$$P_0 = \int_0^R f_S(s) ds, \quad (23)$$

where $f_S(s)$ is the p.d.f of the distance S between two nodes that are independently and uniformly distributed (at random) in a rectangular area of size $a \times b$, where $a \geq b > R$. According to [21], $f_S(s)$ is given by

$$f_S(s) = \frac{4s}{a^2 b^2} \left(\frac{\pi ab}{2} - as - bs + \frac{1}{2s^2} \right) \quad \text{for } 0 \leq s \leq b. \quad (24)$$

Further, when there are $N (\gg 1)$ uniformly distributed nodes in the deployment region, we can expect M nodes in a given CH neighborhood of radius R , where M is given by

$$M = NP_0. \quad (25)$$

Hence, using (23)–(25), we can derive

$$M = \frac{NR^2 \pi}{ab} \left(1 - \frac{4R}{3\pi ab} (a+b) + \frac{R^2}{2\pi ab} \right). \quad (26)$$

Therefore, when (26) is used with (22), we can derive the expected number of CHs. Thus,

$$E[k] = \frac{ab \ln 3}{\sqrt{3}R^2(1 - (4R/3\pi ab)(a+b) + (R^2/2\pi ab))}. \quad (27)$$

As we have already discussed, deriving the CH candidacy broadcasting range R for a desired $E[k]$ is a salient requirement in most applications. Hence rearranging (27), we obtain that

$$\frac{\sqrt{3}}{2\pi ab} R^4 - \frac{4\sqrt{3}(a+b)}{3\pi ab} R^3 + \sqrt{3}R^2 - \frac{ab \ln 3}{E[k]} = 0. \quad (28)$$

By solving (28), we can derive R for a given ad hoc network setup for a rectangular deployment region with the desired number of clusters, $E[k]$ provided that $\pi R^2 \lambda \gg 1$.

5.2. Boundary Effect on $E[k]$ due to a Circular Deployment Region. Let us now derive $E[k]$ for a circular deployment region. We follow the same approach as in the rectangular deployment region case. Let's assume that the ad hoc deployed wireless node network consists of uniform randomly deployed N nodes in the circular deployment region of r radius resulting in $\lambda = N/\pi r^2$.

The expected number of neighboring nodes M in a given CH's CH candidacy broadcasting range R , for a circular deployment area with radius of r is also given by (25). Note that still the P_0 given in (23) is applicable. However $f_S(s)$, that is, the p.d.f of the distance S between two nodes that are independently and uniformly distributed (at random) in a circular area with radius r is given by

$$f_S(s) = \frac{4s}{\pi r^2} \left(\cos^{-1} \left(\frac{s}{2r} \right) - \frac{s}{2r} \sqrt{1 - \left(\frac{s}{2r} \right)^2} \right), \quad (29)$$

for $0 \leq s \leq 2r$,

according to [22]. Hence, we can write the $E[k]$ of a given circular area with radius r as

$$E[k] = \frac{\pi^2 \ln 3}{2\sqrt{3}D(R/r)}, \quad (30)$$

where

$$D(R/r) = 4 \left(\frac{R}{2r} \right)^2 \cos^{-1} \left(\frac{R}{2r} \right) - 3 \left(\frac{R}{2r} \right) \left[1 - \left(\frac{R}{2r} \right)^2 \right]^{1/2} + 2 \left(\frac{R}{2r} \right) \left[1 - \left(\frac{R}{2r} \right)^2 \right]^{3/2} + \sin^{-1} \left(\frac{R}{2r} \right). \quad (31)$$

Thus, we can determine R for a given circular deployment area with radius r for an expected number of clusters $E[k]$ by solving the reordered (30).

Note 1. We derived $E[k]$ assuming that T_i is a random variable. This is true only for the situation where all nodes have equal fitness to be a CH; that is, the residual energies all the nodes are the same. This is in fact true for HEED, ANTCLUST, and EDCR algorithms during initial deployment with the assumption that the sensors are ideal. However in subsequent rounds, T_i would be weighted based on each node's residual energy level at the beginning of the cluster formation. That is, a node with the highest residual energy would be the CH in a given neighborhood.

We know that a node closest to a CH would spend the minimum energy in communication. As a result, it would be the highest energy node in that neighborhood at the beginning of the subsequent CH selection phase. Hence it can be observed that a subsequent round, the CHs would be the nodes closest to the previous CHs. Thus, we can expect on average, the same number of clusters formed in subsequent reclustering rounds as well. As a result, (28) will be valid for all subsequent rounds as well.

In this section, we presented an analytical technique to find the Cluster/CH density of DTDC class of algorithms. Further, we derived the expected number of clusters in a finite area considering the boundary effect. In what follows, we compare the analytical results with simulation experiment results.

6. Simulation Results

In this section, the proposed analytical method to determine the cluster density and expected number of clusters for the DTDC class of algorithms using MATLAB simulations were evaluated. It is already established that the proposed analytical results match the empirical results derived using DMAC algorithm in [18]. For comparison, the simulation results for HEED, ANTCLUST, and EDCR algorithms are presented as well. The results are presented based on the following design scenarios.

- (1) *Design requirement of 20 clusters each with 15 nodes monitoring a square area of $100 \times 100 \text{ m}^2$.* That is, 300 nodes should be deployed in this region. According to (28), the computed broadcasting distance is $R = 19.42 \text{ m}$ to achieve the 20 cluster requirement.
- (2) *Design requirement of 30 clusters each with 20 nodes monitoring a rectangular area of $150 \times 100 \text{ m}^2$.* That is, 600 nodes should be deployed in this region. According to (28) the computed broadcasting distance is $R = 19.11 \text{ m}$ to achieve the 30-cluster requirement.
- (3) *Design requirement of 20 clusters each with 20 nodes monitoring a circular area with radius 200 m.* That is, 400 nodes should be deployed in this region. According to the (30) the computed broadcasting distance is $R = 68.24 \text{ m}$ to achieve the 20 cluster requirement.

The simulation results related to above-described scenarios are given in Table 2. H1, A1, and E1 denotes the results of HEED, ANTCLUST, and EDCR algorithm respectively, for scenario 1 (square area). Similarly, H2, A2 and E2 represents the results for scenario 2 (rectangular area) and H3, A3 and E3 represent the results for scenario 3 (circular area). Note that $E[k]$ denote the desired number of clusters in each case. The average and standard deviation ($AV \pm SD$) of the actual number of clusters ($E[k]^A$) obtained via a large number of different random node deployment simulations corresponding to each scenario has been tabulated. The $E[k]^A$ tabulated in column “Beginning” corresponds to the cluster formation results at the initial deployment stage with

TABLE 2: Applicability of the proposed technique equally among HEED, ANTCLUST, and EDCR algorithms.

Case	$E[k]$	Average number of clusters, $E[k]^A$			
		(AV \pm SD)			
		Beginning	Middle	End	Overall
H1	20	20.4 ± 0.5	19.3 ± 1.2	18.8 ± 1.1	19.5 ± 1.2
A1	20	19.6 ± 0.9	19.4 ± 1.8	18.8 ± 1.5	19.3 ± 1.4
E1	20	19.6 ± 1.5	20.3 ± 0.9	20.4 ± 0.8	20.1 ± 1.1
H2	30	29.1 ± 1.6	28.3 ± 0.8	29.9 ± 0.8	29.1 ± 1.3
A2	30	29.7 ± 1.4	30.1 ± 1.6	28.4 ± 2.0	29.4 ± 1.8
E2	30	29.3 ± 2.0	30.3 ± 1.0	30.2 ± 1.6	29.9 ± 1.6
H3	20	18.9 ± 1.9	20.0 ± 2.2	19.5 ± 1.1	19.5 ± 1.8
A3	20	19.3 ± 1.1	19.6 ± 0.9	18.3 ± 1.7	19.1 ± 1.4
E3	20	19.1 ± 0.7	19.7 ± 1.4	17.9 ± 1.2	18.9 ± 1.3

a fresh set of homogeneous energy nodes, column “End” corresponds to the average number of clusters closer to the end of life of the sensor bed (we used 95% nodes alive as the lifetime measurement [12]), and column “Middle” corresponds to an average number of clusters at a position halfway in between the “Beginning” and “End” scenarios. Further the cumulative average of these three cases is presented in the column “Overall”.

The results given in Table 2 show us that the analytical estimation for R based on $E[k]$ cluster requirement is indeed valid as only a minimal variation of $E[k]$ is seen in all simulation results. These results (based on HEED, ANTCLUST and EDCR algorithms) and independent simulation results of DMAC algorithm (and its corresponding empirical formula) given in [18] affirm the validity and applicability of the proposed analytical technique in determining the cluster density and the expected number of clusters of DTDC class of algorithms.

As it can be seen from Table 2 all major algorithms in the DTDC class respond in a similar manner. Hence without loss of generality the EDCR algorithm can be selected from this class for further analysis. For the analysis 15 different hypothetical node deployment requirements (case) which would cover the applicability of the analytical method with square, rectangular, and circular deployment regions, with different expected number of clusters for a given deployment region, and different expected number of nodes for a given cluster based on the $E[k]$ requirement will be used. These requirements are listed in Table 3. The case number will be used to link the tabulated test results of Table 4 for each of these node deployment requirements. The column given under the heading “Area” presents the dimensions of node-deployed region (e.g., $a \times b$ for a rectangular region and πr^2 for a circular region), while the rest of the columns represents the expected number of clusters $E[k]$, expected number of nodes in a cluster $E[n]$, and the total number of nodes to be deployed in the region $N(= E[k] \times E[n])$. The last column presents the calculated R for each case using either (28) or (30) depending on the shape of the region.

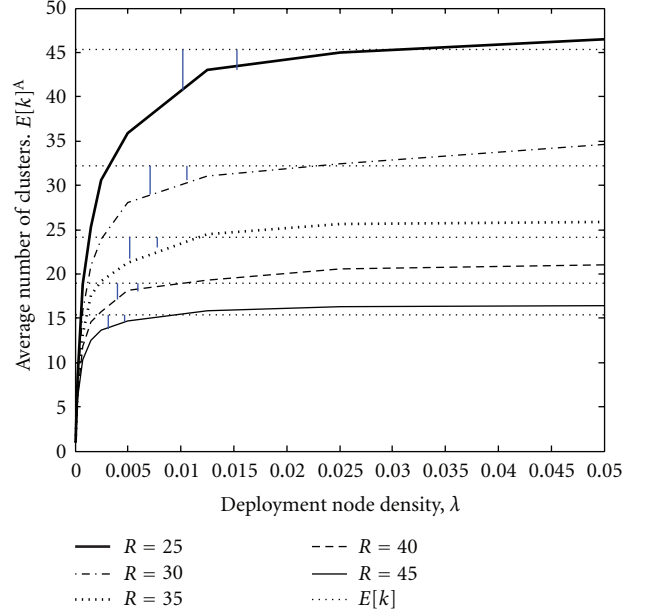
TABLE 3: Network deployment requirements.

Case	Area	$E[k]$	$E[n]$	N	R
1	100×100	20	20	400	19.42
2	100×100	30	20	600	15.57
3	150×100	30	20	600	19.11
4	125×100	40	25	1000	14.93
5	200×200	40	20	800	26.70
6	200×200	30	40	1200	31.15
7	200×200	30	25	750	31.15
8	200×200	30	20	600	31.15
9	200×200	30	15	450	31.15
10	200×200	30	10	300	31.15
11	200×200	10	30	300	57.42
12	$\pi 200^2$	40	15	600	47.04
13	$\pi 200^2$	20	15	300	68.24
14	$\pi 300^2$	20	20	400	102.37
15	$\pi 300^2$	30	30	900	82.24

TABLE 4: Comparison of expected and actual number of clusters of EDCR algorithm.

Case	$E[k]$	Average number of clusters, $E[k]^A$			
		Beginning	Middle	End	Overall
1	20	19.6 ± 1.5	20.3 ± 0.9	20.4 ± 0.8	20.1 ± 1.1
2	30	30.2 ± 1.6	29.5 ± 1.1	30.7 ± 1.3	30.1 ± 1.4
3	30	29.3 ± 2.0	30.3 ± 1.0	30.2 ± 1.6	29.9 ± 1.6
4	40	39.8 ± 2.0	39.6 ± 1.1	41.6 ± 1.1	40.3 ± 1.7
5	40	39.2 ± 1.9	41.0 ± 0.9	40.8 ± 1.4	40.3 ± 1.6
6	30	30.9 ± 1.6	31.1 ± 1.7	32.1 ± 1.2	31.3 ± 1.5
7	30	30.4 ± 1.8	30.9 ± 1.6	31.0 ± 1.7	30.8 ± 1.7
8	30	29.8 ± 1.5	30.3 ± 1.1	30.5 ± 1.5	30.2 ± 1.4
9	30	28.8 ± 1.2	28.3 ± 1.7	28.5 ± 1.5	28.5 ± 1.4
10	30	27.3 ± 1.4	28.0 ± 2.8	28.3 ± 1.4	27.8 ± 2.0
11	10	9.8 ± 0.9	11.0 ± 0.8	10.5 ± 0.9	10.4 ± 1.0
12	40	38.2 ± 1.6	39.6 ± 1.5	39.5 ± 1.7	39.1 ± 1.7
13	20	19.1 ± 0.7	19.7 ± 1.4	17.9 ± 1.2	18.9 ± 1.3
14	20	20.5 ± 1.5	20.7 ± 0.9	18.9 ± 1.6	20.0 ± 1.5
15	30	30.2 ± 1.1	33.1 ± 1.1	32.5 ± 1.0	31.9 ± 1.6

Table 4 shows the simulation results of the deployment requirements listed in Table 3. Table 4 presents the average and standard deviation (AV \pm SD) of the actual number of clusters we observed with the large number of different random node deployments corresponding to each case. The results tabulated in Table 4 indicate that the proposed analytical technique in estimating R for a desired number of clusters $E[k]$ is indeed an accurate method to realize the actual number of clusters. Furthermore, it can be noted that there is minimal variation in $E[k]$ irrespective of the cluster shape (rectangular, square, or circular), desired number

FIGURE 6: “ $E[k]^A$ versus λ ” for different R : $200 \times 200 \text{ m}^2$ square deployment area.

of clusters, and the expected member population in each cluster, provided that all clusters are well populated.

The simulation results presented thus far clearly show the applicability of the proposed analytical technique in estimating the expected number of clusters of the DTDC class of algorithms provided that each cluster is well populated, that is, $\pi R^2 \lambda \gg 1$. In order to identify a minimum threshold for $\pi R^2 \lambda$ or expected number of nodes in a cluster $E[n]$ for a given application requirement, the behavior of curves representing average number of actual clusters, $E[k]^A$ versus different node densities, λ for different CH broadcasting ranges, R can be observed. Figures 6 and 7 present these curves ($E[k]^A$ versus λ) of EDCR algorithm applied for a square deployment region with size $200 \times 200 \text{ m}^2$ and a circular deployment region with radius 100 m, respectively. Both of these graphs consist of $E[k]^A$ versus λ curves for $R = 25, 30, 35, 40$ and 45 . The expected number of clusters, $E[k]$ calculated using (28) and (30) respectively for Figures 6 and 7 and is plotted as a vertical dotted line for each R .

Figures 6 and 7 clearly indicate that all the $E[k]^A$ versus λ curves are asymptotic and close to the expected number of clusters, $E[k]$. The vertical solid error bars marked on each $E[k]$ line shows the 5% (short) and 10% (long) levels below the $E[k]$ at $\pi R^2 \lambda$ are 30 and 20, respectively. It has been already identified that 0.5018 fraction of nodes belonging to any given CH’s broadcasting range R neighborhood (M) to join its cluster in Section 5. Therefore, the proposed analytical technique can be used to determine CH candidacy broadcasting range, R of DTDC class of algorithms with a maximum error of 10% for a required expected number of clusters, $E[k]$, when the expected number of nodes in a cluster, $E[n]$, is more than 10. The number of nodes in

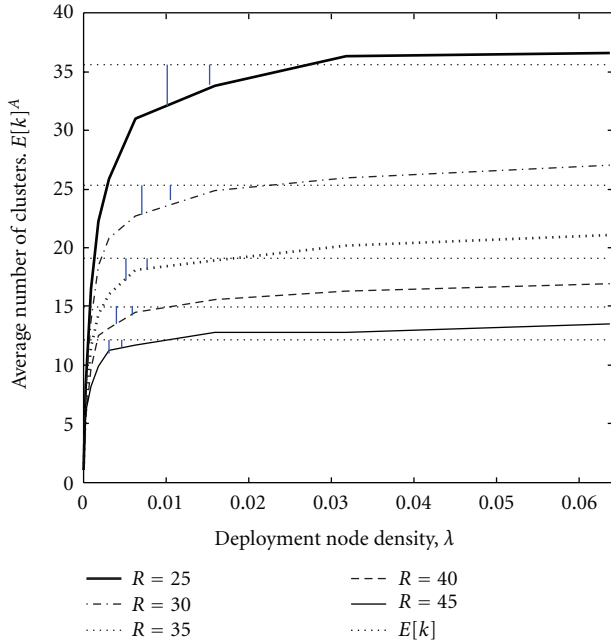


FIGURE 7: “ $E[k]^A$ versus λ ” for different R : 100 m radius circular deployment area.

a cluster is well above this figure in most of the practical applications.

Above-presented simulation results and empirical formula derived based on simulation experiments in [18] affirm the accuracy of using the proposed analytical method in determining R for given expected number of clusters, $E[k]$ of DTDC class of algorithms at the network planning stage.

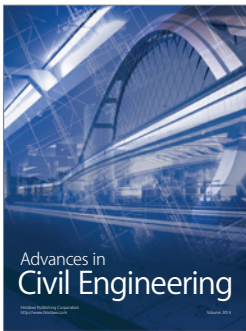
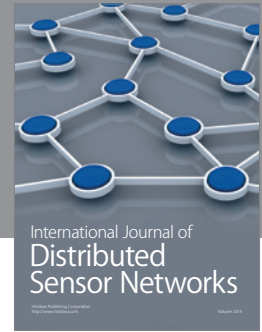
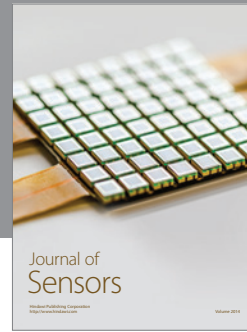
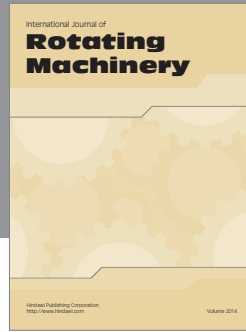
7. Conclusion

Distributed clustering is a popular technique in organizing ad hoc deployed wireless networks including WSNs. We found that clustering algorithms like DMAC, HEED, ANTCLUST, MEDIC, and EDCR can be categorized into the class of DTDC algorithms based on the common underline Dutch Auction principle in CH selection resulting in a similar CH distribution. In this research, we have provided an analytical framework which can be used to derive the cluster density, λ_c , for a given deployment requirement where each cluster is assumed to be well populated. Furthermore, the analysis framework has been extended to include the effects of the boundary resulting from a finite deployment region when computing the expected number of clusters. The proposed analytical technique was verified via simulation experiments, and the results were presented. Further, the empirical formula proposed by Bettstetter in [18] independently verifies the accuracy of the proposed technique and vice versa. The authors feel that this analytical framework can be extended to derive λ_c for any generic situation given by Matérn Type III-dependent thinning point process [20] in future research.

References

- [1] S. Basagni, “Distributed clustering for ad hoc networks,” in *Proceedings of the 4th International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN '99)*, pp. 310–315, Perth/Fremantle, WA, Australia, 1999.
- [2] O. Younis, M. Krunz, and S. Ramasubramanian, “Node clustering in wireless sensor networks: recent developments and deployment challenges,” *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.
- [3] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [4] M. J. Handy, M. Haase, and D. Timmermann, “Low energy adaptive clustering hierarchy with deterministic cluster-head selection,” in *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network*, pp. 368–372, World Scientific Publishing, 2002.
- [5] G. Smaragdakis, I. Matta, and A. Bestavros, “SEP: a stable election protocol for clustered heterogenous wireless sensor networks,” in *Proceedings of the International Workshop on SANPA*, pp. 1–11, Boston, Mass, USA, 2004.
- [6] Y. Wang, Q. Zhao, and D. Zheng, “Energy-driven adaptive clustering data collection protocol in wireless sensor networks,” in *Proceedings of the International Conference on Intelligent Mechatronics and Automation*, pp. 599–604, 2004.
- [7] P. Hebden and A. R. Pearce, “Distributed asynchronous clustering for self-organisation of wireless sensor networks,” in *Proceedings of the 4th International Conference on Intelligent Sensing and Information Processing (ICISIP '06)*, pp. 37–42, 2006.
- [8] Q. Wang, H. Hassanein, and G. Takahara, “Stochastic modeling of distributed, dynamic, randomized clustering protocols for wireless sensor networks,” in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW '04)*, pp. 456–463, 2004.
- [9] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [10] J. Kamimura, N. Wakamiya, and M. Murata, “Energy-efficient clustering method for data gathering in sensor networks,” in *Proceedings of the 1st Workshop on Broadband Advanced Sensor Networks*, San Jose, Calif, USA, 2004.
- [11] L. Zhao and Q. Liang, “Medium-contention based energy-efficient distributed clustering (MEDIC) for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 3, no. 4, pp. 347–369, 2007.
- [12] S. Gamwarige and C. Kulasekera, “An energy efficient distributed clustering algorithm for ad hoc deployed wireless sensor networks in building monitoring applications,” *Electronic Journal of Structural Engineering*, pp. 11–27, 2009.
- [13] S. Gamwarige and C. Kulasekera, “A cluster based energy balancing strategy to improve wireless sensor network lifetime,” in *Proceedings of the 2nd International Conference on Industrial and Information Systems (ICIIS '07)*, pp. 403–408, 2007.
- [14] T. Sun, L.-J. Chen, C.-C. Han, and M. Gerla, “Improving data reliability via exploiting redundancy in sensor networks,” Tech. Rep. csd-tr no. 040037, UCLA Computer Science Department, Los Angeles, Calif, USA, 2004.
- [15] L.-C. Wang, C.-W. Wang, and C.-M. Liu, “Optimal number of clusters in dense wireless sensor networks: a cross-layer

- approach,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 2, pp. 966–976, 2009.
- [16] T. J. Chan, C. M. Chen, Y. F. Huang, J. Y. Lin, and T. R. Chen, “Optimal cluster number selection in ad-hoc wireless sensor networks,” *WSEAS Transactions on Communications*, vol. 7, no. 8, pp. 837–846, 2008.
- [17] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, “A minimum cost heterogeneous sensor network with a lifetime constraint,” *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 4–14, 2005.
- [18] C. Bettstetter, “The cluster density of a distributed clustering algorithm in ad hoc networks,” in *Proceedings of the IEEE International Conference on Communications*, pp. 4336–4340, June 2004.
- [19] B. Matérn, *Spatial Variation*, vol. 36, Springer, New York, NY, USA, 2nd edition, 1986.
- [20] M. L. Huber and R. L. Wolpert, “Likelihood-based inference for Matern Type III re-pulsive point processes,” Under Invited Revision in Applied Probability Trust, 2009, <http://ftp.stat.duke.edu/WorkingPapers/08-27.pdf>.
- [21] B. Gosh, “Random distances within a rectangle and between two rectangles,” *Bulletin of the Calcutta Mathematical Society*, vol. 43, pp. 17–24, 1951.
- [22] H. C. Tuckwell, *Elementary Applications of Probability Theory*, Chapman & Hall, London, UK, 2nd edition, 1995.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

