Minimizing Redundant Sensing Data Transmissions in Energy-Harvesting Sensor Networks via Exploring Spatial Data Correlations

Zhenjie Guo, Jian Peng, Wenzheng Xu, Member, IEEE, Weifa Liang, Senior Member, IEEE, Weigang Wu, Member, IEEE, Zichuan Xu, Member, IEEE, Bing Guo, and Yue Ivan Wu, Member, IEEE

Abstract—Energy harvesting rates of sensors in renewable (e.g., solar energy) wireless sensor networks are not only lower than their energy consumption rates but also temporally varying. Existing studies exploited spatial data correlations among sensors to reduce their energy consumption rates, but the data correlations mean that the sensing data of nearby sensors have high similarities. They assumed that the sensing data of nearby sensors are very likely to highly correlated. They adopted a coarse-grained spatial-correlation model, in which sensors are partitioned into different clusters such that the sensors in the same cluster have high data similarities with each other. Then, only the sensor with the maximum residual energy in each cluster sends its sensing data, while the other sensors do not. We, however, notice that the data similarities among nearby sensors in real-sensor networks may vary significantly, i.e., ranging from very similar to not similar at all. Since the existing algorithms require that the sensors in the same cluster have high data similarities with each other, the sensors in a network may be partitioned into many clusters and each cluster consists of only a few sensors, where two nearby sensors belong to two different clusters if the sensing data of the two sensors are not highly correlated. Therefore, in the existing studies, many sensors have to send all their data as there are many clusters. Unlike the existing studies, in this article, we first propose a fine-grained spatial correlation model, in which sensors are partitioned into only a few clusters and each cluster consists of many sensors. Then, each cluster master sensor sends all its data to the sink, while the majority of other sensors in the cluster transmit only their nonredundant data, thereby significantly saving sensor energy consumptions. We formulate a novel sensor clustering problem under the proposed model, which is to partition sensors into different clusters and choose a representative sensor for each cluster such that the amount of suppressed redundant data transmissions is maximized. We propose a randomized $(0.5 - \epsilon)$-approximation algorithm for the clustering problem, where $\epsilon$ is a given constant with $0 < \epsilon \leq 0.5$. To further reduce sensor energy consumption, we consider temporal data correlations, where the sensing data by a sensor in a short period are likely to be highly correlated. We investigate a data utility maximization problem that allocates sensor data rates and routing so that the accumulative utility of both spatially and temporally correlated data received by the sink is maximized. We devise a near-optimal algorithm for the problem. We finally evaluate the performance of the proposed algorithms through experiments. The experimental results show that the proposed algorithms are very promising.

Index Terms—Approximation algorithm, energy-harvesting sensor networks, redundant sensing data, spatial data correlations.

I. INTRODUCTION

WIRELESS sensor networks (WSNs) have wide applications, including industrial automation systems monitoring, environmental monitoring, smart grid network monitoring, Internet of Things (IoT), etc. [9], [11]. In conventional WSNs, sensors are powered by energy-limited batteries, the network lifetime thus is restricted by sensor batteries, as sensors will not function when they run out of energy. To address the network lifetime issue, some pioneering researchers proposed to enable sensors to harvest energy from their surrounding environment, such as solar energy, wind energy, thermal energy, etc. [4], [28]. It is well known that data transmission and reception between sensors are very energy-consuming [19], while the energy harvesting rate of a sensor usually is lower than its energy consumption rate, since its attached energy harvesting component cannot be too large, due to practical restrictions on
Fig. 1. Energy harvesting profiles in a renewable sensor network. (a) Sensor network. (b) Energy harvesting rates of sensors $v_1$ and $v_2$ at different time from [37]. (c) Energy harvesting rates of sensor $v_3$ at different time points from [37].

![Sensor Network Diagram](image1)

![Energy Harvesting Profiles](image2)

Fig. 2. Temperature sensing data by three sensors $v_1$, $v_2$, and $v_3$ from 2:00 to 20:00 in a day.

sensor size or its cost [9], [11]. In addition, the energy harvesting rate of a sensor is temporally and spatially varying [37]. For example, Fig. 1(a) shows a sensor network, which consists of three sensors $v_1$, $v_2$, and $v_3$ and a sink for temperature monitoring. Fig. 1(b) plots the energy harvesting rates of sensors $v_1$ and $v_2$ at different time points in a day [6], [37]. In contrast, Fig. 1(c) plots the energy harvesting rate of sensor $v_3$ in the afternoon is very low since it is within the shadow area of a nearby building. Due to the variability of the harvested energy in renewable sensor networks, one fundamental question is how to make full use of the harvested energy such that the amount of data collected from sensors is maximized.

It is recognized that sensing data generated by nearby sensors exhibit high similarities, which is referred to as spatial data correlations [8], [29], [35]. For example, Fig. 2 shows the temperature sensing data by three sensors $v_1$, $v_2$, and $v_3$ from 2:00 to 20:00 in a day, and Table I keeps their detailed records [16]. It can be seen from Fig. 2 that the temperature data by sensors $v_1$ and $v_2$ are very similar with each other, while they have some deviations only at time 4:00. In contrast, the sensing data of $v_3$ after 14:00 have large differences from that of both $v_1$ and $v_2$, where the difference of the temperature data from two sensors at a time slot is referred to be large if the difference is larger than a given threshold $\gamma$, e.g., $\gamma = 0.5$. Following the existing studies [16], [22], the data similarities among the three sensors can be calculated as follows. The data similarity $c(1, 2)$ between sensors $v_1$ and $v_2$ is $[(10 - 1)/10] = 0.9$, since their data have a large difference only at time point 4:00 among the 10 time points in Table I. The data similarities $c(1, 3)$ between sensors $v_1$ and $v_3$, and $c(2, 3)$ between $v_2$ and $v_3$ can be calculated similarly, which are $[(10 - 4)/10] = 0.6$ and $[(10 - 5)/10] = 0.5$, respectively.

Existing studies proposed to explore the high spatial data correlations among sensors to reduce the transmission of redundant sensing data, thereby saving the energy consumption of energy-limited sensors [22], [32], [35]. They partition sensors into different clusters such that the sensors in each cluster have high spatial data correlations with each other. The sensor with the maximum residual energy in each cluster transmits its sensing data to the sink (acting as a representative node), while the others do not transmit their sensing data. For example, the three sensors $v_1$, $v_2$, and $v_3$ in Fig. 3(a) will be partitioned into two clusters $C_1 = \{v_1, v_2\}$ and $C_2 = \{v_3\}$, due to the high data similarity between $v_1$ and $v_2$. Then, sensor $v_1$ sends its sensing data to the sink, while sensor $v_2$ in $C_1$ does not. Also, sensor $v_3$ transmits its data to the sink, too. Assume that the data generating rate of each of the three sensors is 1 kb/s. Then, the throughput received by the sink is 1 kb/s + 1 kb/s = 2 kb/s, which is less than the accumulative data generating rate of the three sensors, i.e., 1 kb/s \times 3 = 3 kb/s, thereby saving the energy consumption of sensor $v_2$. On the other hand, the data collected by the sink still maintain high quality.

Although existing studies have conducted excellent work on the exploitation of spatial data correlations, such exploitation is in the coarse-grained level, i.e., each sensor either transmits its sensing data to the sink or not. In this article, we observe

![Temperature Data Monitored by Sensors](image3)

**Table I**

<table>
<thead>
<tr>
<th>Time</th>
<th>data of $v_1$</th>
<th>data of $v_2$</th>
<th>data of $v_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:00</td>
<td>10</td>
<td>10</td>
<td>9.8</td>
</tr>
<tr>
<td>4:00</td>
<td>9.3</td>
<td>9.8</td>
<td>9.2</td>
</tr>
<tr>
<td>6:00</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>8:00</td>
<td>9.1</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10:00</td>
<td>11</td>
<td>11</td>
<td>10.9</td>
</tr>
<tr>
<td>12:00</td>
<td>12.6</td>
<td>12.8</td>
<td>12.5</td>
</tr>
<tr>
<td>14:00</td>
<td>14</td>
<td>14.2</td>
<td>13.2</td>
</tr>
<tr>
<td>16:00</td>
<td>13.3</td>
<td>13.5</td>
<td>12.5</td>
</tr>
<tr>
<td>18:00</td>
<td>12.6</td>
<td>12.6</td>
<td>11</td>
</tr>
<tr>
<td>20:00</td>
<td>12</td>
<td>12</td>
<td>10</td>
</tr>
</tbody>
</table>
that although some sensor \( v \) may not have a very high spatial correlation with its nearby sensor \( u \), their sensing data indeed have some similarity to a certain degree. We make statistics of real spatial temperature data correlations between nearby sensors in a sensor network deployed by the Intel Berkeley Research laboratory [12], where 54 sensors are deployed in the network. Fig. 4 shows the distribution of the spatial data correlations, from which it can be seen that the spatial data correlations vary from very small to very large.

In addition, Fig. 3(a) shows that sensor \( v_3 \) has a data similarity \( c(1,3) = 0.6 \) with sensor \( v_1 \). Following the existing studies [22], [32], [35], sensor \( v_3 \) should transmit its sensing data to the sink, since the data similarity \( c(1,3) \) between \( v_1 \) and \( v_3 \) is not high enough, e.g., \( c(1,3) = 0 \) should be no less than 0.9, while \( c(1,3) = 0.6 \).

In contrast, in this work, we exploit a fine-grained spatial correlation model, in which some sensors send all their sensing data, while the others transmit only nonredundant data. For example, in Fig. 3(b), sensor \( v_1 \) sends all its data back to the sink, while both \( v_2 \) and \( v_3 \) send their nonredundant data with respect to the data by \( v_1 \). Table II shows the data sent back to the sink by sensors \( v_1 \), \( v_2 \), and \( v_3 \), respectively. It can be seen that the total throughput received by the sink is 1.5 kb/s (\( 1 \text{ kb/s} + 0.1 \text{ kb/s} + 0.4 \text{ kb/s} \)), which is less than that by the existing studies, i.e., 1.5 kb/s < 2 kb/s. On the other hand, the data sent by \( v_1 \), \( v_2 \), and \( v_3 \) under this fine-grained spatial-correlation model still maintain a similar high quality against the data by the existing studies under the coarse-grained spatial-correlation model.

We consider an important clustering problem in the fine-grained model, which is to partition sensors into different clusters, and one sensor in each cluster is chosen to serve as the master sensor, while the others serve as slave sensors. The master sensor in each cluster sends its data to the sink, while the other slave sensors in the cluster transmit only their nonredundant data. The objective is to maximize the amount of suppressed transmissions of redundant data for the entire network.

The clustering problem is very challenging since each chosen master sensor in a cluster should not only have high spatial data correlations with its slave sensors but also have a large amount of available energy. Otherwise, even if the chosen master sensor with a little energy has high spatial data correlations with the data by its slave sensors, it can only generate representative data for a short period. On the other hand, if the chosen master sensor with a large amount of available energy has low spatial data correlations with the data by its slave sensors, then the slave sensors have to send more nonredundant data by themselves, thereby consuming more energy. In this article, we address the challenge by proposing a novel \( (0.5 - \epsilon) \)-approximation algorithm for the clustering problem, where \( \epsilon \) is a given constant with \( 0 < \epsilon \leq 0.5 \).

In addition to the spatial data correlation, some researchers also exploited temporal data correlations, where sensing data by a sensor in a short period are also highly correlated [6], [37]. For example, it can be seen that the temperature
data by vi at 6:00 and 8:00 in Table I are similar, i.e., 9 °C and 9.1 °C, respectively.

To further enhance energy savings of sensors, we also explore the temporal data correlations. We refer to this problem as the data utility maximization problem, which is to allocate a data rate for each sensor and the routing among sensors for sending the data to the sink such that the utility of the spatially temporally correlated sensing data received by the sink is maximized, subject to the time-varying energy constraints on sensors.

The main contributions of this article are summarized as follows.

1) Unlike the existing studies that adopted the coarse-grained spatial-correlation model, in which each sensor either sends its data to the sink or not, in this article, we introduce a novel fine-grained spatial-correlation model, in which only a small portion of sensors send their representative data and the other majority sensors transmit their nonredundant data, thereby reducing redundant data transmissions within the network.

2) We first consider a clustering problem under the proposed fine-grained spatial-correlation model, which is to partition sensors into different clusters and choose a master sensor for each cluster such that the amount of suppressed redundant data transmissions is maximized. We propose a novel approximation algorithm for the problem, which delivers a \((0.5 - \epsilon)\)-approximate solution with high probability, where \(\epsilon\) is a given constant with \(0 < \epsilon \leq 0.5\).

3) We then devise an efficient algorithm for the data utility maximization problem, by allocating a near-optimal data rate for each sensor and routing among sensors such that the utility of the spatially temporally correlated data received by the sink is maximized.

4) We finally evaluate the performance of the proposed algorithms, using real data sets. The experimental results show that the amount of suppressed redundant data transmissions and the accumulative utility by the proposed algorithms are approximately 50% and 17% larger than those by the existing algorithms, respectively.

The remainder of the article is organized as follows. Section II introduces the network model and defines the problems. Section III proposes an approximation algorithm for the clustering problem. Section IV devises a near-optimal algorithm for the data utility maximization problem. Section V evaluates the performance of the proposed algorithms. Section VI reviews related work and Section VII concludes the article.

II. Preliminaries

In this section, we introduce the network model, energy model, spatial data correlation model, and define the problems precisely.

A. Network Model

Consider an energy harvesting sensor network \(G = (V \cup \{s\}, E)\), which consists of \(n\) sensors \(v_1, v_2, \ldots, v_n\) and a sink \(s\), where \(n = |V|\) and \(v_i \in V\) with \(1 \leq i \leq n\). The sensors are randomly deployed in a 2-D space and the sink \(s\) is located at the center of the area for sensing data collection. We assume that each sensor \(v_i\) in \(V\) is powered by a rechargeable battery \(B_i\), which can harvest energy from its surrounding environment, such as solar energy, wind energy, etc. There is an edge in \(E\) between two sensors or between one sensor and the sink if they are within the transmission range of each other. Let \(m = |E|\). Denote by \(N(v_i)\) the set of neighbors of sensor \(v_i\) in \(G\), i.e., \(N(v_i) = \{v_j \mid v_j \in V \cup \{s\}, (v_i, v_j) \in E\}\).

We consider the data collection in the network \(G\) for a monitoring period of \(T\) (e.g., \(T = 24\) h) that is divided into equal time slots \(1, 2, \ldots, T\). Each time slot lasts for a duration of \(\tau\) (e.g., \(\tau = 1\) h). The sensing rate of each sensor \(v_i\) is \(R_i\), but it sends a portion of its sensing data at a rate of \(r_i(t)\) at time slot \(t\) since there are some redundant data in the sensing data and these data are not sent to the sink. Then, \(0 \leq r_i(t) \leq R_i\). Denote by \(f_i(t)\) and \(f_j(t)\) the data transmission rates from sensors \(v_i\) to \(v_j\), and from \(v_j\) to \(v_i\), at time slot \(t\), respectively, where \((v_i, v_j) \in E\). Notice that the data transmitted from sensors \(v_i\) to \(v_j\) include the data from \(v_j\) itself and the data received from other sensors.

B. Energy Model

It is well known that each sensor \(v_i\) consumes energy on data sensing, data transmission, and data reception. Denote by \(P(v_i, t)\) the total energy consumption of sensor \(v_i\) at time slot \(t\), then

\[
P(v_i, t) = P_S(v_i, t) + P_T(v_i, t) + P_R(v_i, t)
\]

(1)

where \(P_S(v_i, t)\), \(P_T(v_i, t)\), and \(P_R(v_i, t)\) are the amounts of energy consumption of \(v_i\) for data sensing, data transmission, and reception at time slot \(t\), respectively. The values of \(P_S(v_i, t)\), \(P_T(v_i, t)\), and \(P_R(v_i, t)\) can be precisely calculated as follows. By adopting a real energy consumption model from [14], we have

\[
P_S(v_i, t) = \lambda \cdot R_i \cdot \tau
\]

(2)

\[
P_T(v_i, t) = \sum_{v_j \in N(v_i)} f_{ij}(t) \cdot (\beta_1 + \beta_2 d_{ij}^2) \cdot \tau
\]

(3)

\[
P_R(v_i, t) = \sum_{v_j \in N(v_i)} f_{ji}(t) \cdot e_R \cdot \tau
\]

(4)

where \(\lambda, R_i, \text{ and } \tau\) in (2) are the energy consumption for sensing per unit data, the sensing data rate of sensor \(v_i\), and the duration of a time slot, respectively, \(f_{ij}(t)\) and \((\beta_1 + \beta_2 d_{ij}^2)\) in (3) are the data transmission rate from \(v_i\) to \(v_j\), and the energy consumption from \(v_i\) to \(v_j\), \(d_{ij}\) is the Euclidean distance between sensors \(v_i\) and \(v_j\), and \(f_{ji}(t)\) and \(e_R\) in (4) are the data transmission rate from \(v_j\) to \(v_i\) and the energy consumption for receiving per unit data, respectively. Notice that the values of \(\lambda, \tau, \beta_1, \beta_2, \alpha, \text{ and } e_R\) are constants [14].

Sensors can harvest energy from their surrounding environments, such as solar energy. Denote by \(H(v_i, t)\) the amount of harvested energy in each sensor \(v_i\) at each time slot \(t\). Also, denote by \(RE(v_i, t - 1)\) and \(RE(v_i, t)\) the amounts of residual energy of sensor \(v_i\) after time slots \(t - 1\) and \(t\), respectively. Then

\[
RE(v_i, t) = \min\{RE(v_i, t - 1) + H(v_i, t) - P(v_i, t), B_i\}
\]

(5)
where $H(v_i, d, t)$ and $P(v_i, t)$ are the amounts of generated and consumed energy by sensor $v_i$ at time slot $t$, and $B_i$ is the battery capacity of sensor $v_i$.

C. Spatial Data Correlation Model

It is recognized that a group of sensors close with each other may have similar sensing data at the same time. For example, in a WSN for temperature monitoring, the temperature data monitored by two nearby sensors usually are highly correlated. In this case, we may just collect temperature data from one of them, thereby saving the energy of the other for future data collection.

We measure the spatial data correlation between two sensors $v_i$ and $v_j$ by their data similarity for a given long period $P$, e.g., one month. Let $r_{i1}, r_{i2}, \ldots, r_{ik}$ be the sensing data of sensor $v_i$ at time slots $t_{i1}, t_{i2}, \ldots, t_{ik}$ with $t_{i1} \leq t_{i2} \leq \cdots \leq t_{ik}$, and let $r_{j1}, r_{j2}, \ldots, r_{jk}$ be the sensing data of sensor $v_j$ at time slots $t_{j1}, t_{j2}, \ldots, t_{jk}$ with $t_{j1} \leq t_{j2} \leq \cdots \leq t_{jk}$. For example, Table III shows that sensor $v_1$ generates data at time slots 1, 2, 6, and 8, but no data at time slots 3, 4, 5, and 7, while sensor $v_2$ generates data at time slots 1, 3, 6, and 8, but no data at time slots 2, 4, and 5.

It can be seen from Table III that there are some time slots that one sensor generates data, whereas the other does not, e.g., sensor $v_2$ has data at time slot $t = 3$, while $v_1$ does not. In order to measure the spatial data correlation of $v_i$ and $v_j$, we need to estimate the missed data such that for each time slot $t$, either both $v_i$ and $v_j$ have data or none of them have data.

Let $r_{i,t}$ and $r_{j,t}$ be the predicted data of sensors $v_i$ and $v_j$ at time slot $t$, respectively. There are three cases to be considered.

*Case 1:* Both sensors $v_i$ and $v_j$ have data $r_{i,t}$ and $r_{j,t}$, then $r_{i,t} = r_{i,t}$ and $r_{j,t} = r_{j,t}$.

*Case 2:* None of $v_i$ and $v_j$ have data at time slot $t$. Then, we do not estimate their data.

*Case 3:* Sensor $v_i$ has data at time slot $t$, but sensor $v_j$ does not. Then, $r_{i,t} = r_{i,t}$ and we estimate the value of $r_{j,t}$ as follows. We can adopt a quadratic function or a cubic function to curve fit the missed data. For example, consider $v_2$ has data $r_{2,3} = 12$ at time slot $t = 3$, whereas $v_1$ does not. We estimate $r_{1,3}$, by noticing that $v_1$ has data at time slots $t = 1$, $t = 2$, and $t = 6$. Then, we can obtain a quadratic function $y = ax^2 + bx + c$ that passes the three points $(1, 15)$, $(2, 13.1)$, and $(6, 8)$, where $a = 0.125$, $b = -2.275$, and $c = 17.15$, i.e., $y = 0.125x^2 - 2.275x + 17.15$. The missed data of $v_1$ at $t = 3$ are estimated as $r_{1,3} = 0.125 \cdot 3^2 - 2.275 \cdot 3 + 17.15 = 11.45$.

Table IV shows the estimated data of both sensors $v_1$ and $v_2$.

We measure the spatial data correlation between two sensors $v_i$ and $v_j$ by their data similarity for a given long period $P$, e.g., one month. Let $r_{i1}, r_{i2}, \ldots, r_{iP}$ be the sensing data of sensor $v_i$ at time slots 1, 2, $P$, respectively. Also, let $r_{j1}, r_{j2}, \ldots, r_{jP}$ be the sensing data of sensor $v_j$ at these $P$ time slots. Denote by $K_{ij}$ the number of time slots that the difference of the data between $v_i$ and $v_j$ is no greater than a given small threshold $\gamma$, i.e., $K_{ij} = \sum_{t=1}^{P} I(|r_{i,t} - r_{j,t}|)$, where $I(|r_{i,t} - r_{j,t}|) = 1$ if $|r_{i,t} - r_{j,t}| \leq \gamma$; otherwise, $I(|r_{i,t} - r_{j,t}|) = 0$. Table IV indicates that $K_{ij} = 5$ when $\gamma = 0.5$.

Denote by $P'$ the number of time slots in period $P$ such that both sensors $v_i$ and $v_j$ have data. For example, Table IV shows that $P' = 6$. The spatial data correlation $c(i, j)$ between sensors $v_i$ and $v_j$ is defined as the ratio of $K_{ij}$ to $P'$, i.e.,

$$c(i, j) = \frac{K_{ij}}{P'}. \quad (6)$$

For example, the spatial data correlation $c(1, 2)$ between $v_1$ and $v_2$ in Table IV then is $c(1, 2) = (5/6)$.

D. Problem Definitions

In an energy-harvesting WSN, it is well known that the energy harvesting rate of each sensor is lower than its energy consumption rate [20]. In order to make full use of limited harvested energy, we can reduce the energy consumption of sensors, by enabling only energy-sufficient sensors to generate representative data for their nearby sensors with less energy due to their data similarities, while the nearby sensors send only a small amount of their nonredundant data to the sink. By doing so, the quality of the collected data may be compromised. To maximize the quality of collected data in energy harvesting sensor networks, we here consider two novel problems. The first one is to partition sensors into different clusters and a representative sensor is chosen for each cluster. The second one is to allocate sensor data rates and routing, such that the accumulative utility of both spatially correlated and temporally correlated data received by the sink is maximized. We formally define the two problems as follows.

1) Redundant Data Suppression Maximization Problem: Given an energy-harvesting WSN $G = (V, E)$, and spatial data correlations $c(v_i, v_j)$ between sensors, we consider the problem of partitioning the sensors in $V$ into $k$ clusters $C_1, C_2, \ldots, C_k$, where $H(v_i, d, t)$ and $P(v_i, t)$ are the amounts of generated and consumed energy by sensor $v_i$ at time slot $t$, and $B_i$ is the battery capacity of sensor $v_i$. The estimated data of both sensors $v_1$ and $v_2$.
where $k$ is not given and is determined by the data correlation relationships. One sensor $u_k$ in each $C_i$ is chosen as the master sensor, and the rest sensors in $C_i \setminus \{u_k\}$ are the slave sensors of $u_k$, where each slave sensor $v_j$ in $C_i$ is a neighbor of the master sensor $u_k$ (i.e., $v_j \in N(u_k)$). Then, $C_i \cap C_j = \emptyset$ if $i \neq j$, $\cup_{i=1}^{k} C_i = V$. For example, Fig. 5 shows that six sensors in a sensor network are partitioned into $k = 2$ clusters $C_1$ and $C_2$, where $C_1 = \{v_1, v_2, v_3\}$, $C_2 = \{v_4, v_5, v_6\}$, and $v_2$ and $v_6$ are the master sensors of $C_1$ and $C_2$, respectively.

In the process of data sensing, for each cluster $C_i$ with its master sensor $u_i$, assume that $u_i$ generates data at a rate of $r_i(t)$ at time slot $t$. For each slave sensor $v_j \in C_i \setminus \{u_i\}$, $v_j$ can generate at a lower data rate $r_j(t)$ due to the data similarity $c(i,j)$ between $v_j$ and its master sensor $u_i$.

For each cluster $C_i$ with its master sensor $u_i$, the total energy budget of $u_i$ in period $T$ is
\[ E^T_i = RE(u_i, 0) + \sum_{t=1}^{T} H(u_i, t) \tag{7} \]
where $RE(u_i, 0)$ is the residual energy at the beginning of the monitoring period and $H(u_i, t)$ is the amount of harvested energy at time slot $t$ with $1 \leq t \leq T$. On the other hand, the per sensor energy-consuming rate $\rho_i$ of sensor $u_i$ can be calculated by following the realistic energy model from [14, eq. (7), p. 8], which incorporates the energy consumptions for sensing, data transmission, and data reception. It can be seen that the residual lifetime of sensor $u_i$ is $l_i = (E^T_i / \rho_i)$. The maximum amount of data $D^\text{max}_i$ that can be generated by $u_i$ in period $T$ then is
\[ D^\text{max}_i = R_i \cdot l_i \tag{8} \]
where $R_i$ is the sensing data rate of $u_i$.

Due to the data similarity $c(i,j)$ between the master sensor $u_i$ of $C_i$ and each slave sensor $v_j$ in $C_i \setminus \{u_i\}$, the amount of suppressed redundant data transmissions by $v_j$ due to the generated data $D^\text{max}_i$ by $u_i$ is $D^\text{max}_i \cdot c(i,j)$, where $0 \leq c(i,j) \leq 1$. The total amount of suppressed redundant data transmissions by sensors in $C_i \setminus \{u_i\}$ then is $\sum_{v_j \in C_i \setminus \{u_i\}} D^\text{max}_i \cdot c(i,j)$. Given clusters $C_1, C_2, \ldots, C_k$ with their master sensors $u_1, u_2, \ldots, u_k$, it can be seen that the total amount of suppressed redundant data transmissions in the network is
\[ \sum_{i=1}^{k} \sum_{v_j \in C_i \setminus \{u_i\}} D^\text{max}_i \cdot c(i,j). \tag{9} \]

In this article, we consider a redundant data suppression maximization problem, which is to partition $n$ sensors into $k$ disjoint clusters $C_1, C_2, \ldots, C_k$ and find a master sensor $u_i$ for each cluster $C_i$, such that the total amount of suppressed redundant data transmissions is maximized, where $k$ is not given and it is dynamically determined by an algorithm for this problem. Then, the problem is
\[ \max \sum_{i=1}^{k} \sum_{v_j \in C_i \setminus \{u_i\}} D^\text{max}_i \cdot c(i,j). \tag{10} \]

2) Utility Maximization Problem: To further enhance the energy saving of sensors, we also explore the temporal data correlations. We term this problem as the data utility maximization problem, which is to allocate a data rate for each sensor and the routing among sensors for sending generated data to the sink such that the utility of the spatially temporally correlated sensing data received by the sink is maximized, subject to the temporally varying energy constraints on sensors.

Recall that $r_i(t)$ is the data rate of each sensor $v_i$ in $V$ at each time slot $t$. Also, $f_j(t)$ is the data transmission rate from sensor $v_i$ to $v_j$ at time slot $t$. For each cluster $C_i$ with its master sensor $u_i$, it can be seen that the amounts of data sent by $u_i$ and each slave sensor $v_j$ in $C_i$ for the monitoring period $D_i = \sum_{t=1}^{T} r_i(t) \cdot \tau$ and $D_j = \sum_{t=1}^{T} r_j(t) \cdot \tau$, respectively, where $\tau$ is the duration of each time slot. On the other hand, due to the spatial correlation $c(i,j)$ between the master sensor $u_i$ and slave sensor $v_j$, the amount of obtained data $D_{ij}$ by $v_j$ is the sum of the amount of data sent by $v_i$ itself and the amount of similar data between $v_j$ and its master sensor $u_i$, i.e., $D_{ij} = D_i + D_j \cdot c(i,j)$.

We introduce a utility function $U(\cdot)$ to measure the quality of collected data from each sensor, and we aim to maximize the accumulative utility of the spatially temporally correlated data collected from sensors. Let $U(\cdot)$ be a network utility function, which is defined as an increasing, twice-differentiable, and strictly concave function. For example, $U(x) = \log_2 (x+1)$, where $x$ is a nonnegative number. The utility function can characterize the diminishing return property, i.e., collecting an additional amount of data from a sensor results in a smaller increase in the nonredundant data. This encourages to collect data from many sensors, rather than from a few sensors. For each cluster $C_i$, the utilities of the data from the master sensor $u_i$ and each slave sensor $v_j \in C_i \setminus \{u_i\}$ are $U(D_i)$ and $U(D_{ij})$, respectively. Therefore, the accumulative utility of the data received by the sink is
\[ \sum_{i=1}^{k} \left( U(D_i) + \sum_{v_j \in C_i \setminus \{u_i\}} U(D_{ij}) \right). \tag{11} \]

We consider a novel utility maximization problem, which is to allocate a data rate $r_i(t)$ for each sensor and routing $f_{ji}(t)$ among sensors such that the accumulative utility of the spatially temporally correlated data received by the sink is maximized, subject to the energy constraints on sensors.

Notice that when there are no spatial correlations among sensors, i.e., $c(i,j) = 0$, each sensor then forms a cluster
by itself, and the objective function in (11) degenerates to 
\[ \sum_{i=1}^{n} U(D_i) \], which is identical to the existing studies that took only temporal correlations into account [6].

III. APPROXIMATION ALGORITHM FOR THE REDUNDANT DATA SUPPRESSION MAXIMIZATION PROBLEM

In this section, we deal with the redundant data suppression maximization problem, by proposing a novel approximation algorithm.

A. Algorithm

Recall that the redundant data suppression maximization problem is to partition \( n \) sensors \( v_1, v_2, \ldots, v_n \) into \( k \) clusters \( C_1, C_2, \ldots, C_k \), and find a master sensor for each cluster \( C_i \) such that the total amount of suppressed redundant data transmissions by slave sensors is maximized.

The basic idea behind the proposed approximation algorithm is that we reduce the problem to another equivalent problem in a novel way. Given any two sensors \( v_i \) and \( v_j \), denote by \( p_{ij} \) the amount of suppressed redundant data transmissions if \( v_i \) is the master node of \( v_j \), i.e., \( p_{ij} = D_{ij}^{\text{max}} \cdot c(i,j) \), where \( D_{ij}^{\text{max}} \) is the maximum amount of generated data by \( v_i \) and \( c(i,j) \) is the data similarity between \( v_i \) and \( v_j \). Especially, \( p_{ij} = 0 \) if \( v_j \) is not a neighbor of \( v_i \) or \( i = j \).

Denote by \( M \) the set of master sensors in the network. Then, \( M \subseteq V \), where \( V = \{v_1, v_2, \ldots, v_n\} \). For each slave sensor \( v_j \in V \setminus M \), it can be seen that \( v_j \) may be a slave of multiple master nodes. In order to maximize the amount of suppressed redundant data transmissions by \( v_j \), its master sensor should be a sensor \( u \in M \) such that \( p_{uj} = \max_{v_i \in M} p_{ij} \). Then, the amount of suppressed redundant data transmissions by slave sensors in \( V \setminus M \) can be calculated as \( f(M) = \sum_{v_j \in V \setminus M} \max_{v_i \in M} p_{ij} \). Therefore, the redundant data suppression maximization problem is equivalent to find a set \( M \subseteq V \) of master sensors such that

\[
\max_{M \subseteq V} f(M) = \max_{M \subseteq V} \left\{ \sum_{v_j \in V \setminus M} \max_{v_i \in M} p_{ij} \right\} .
\] (12)

We later will show that function \( f(M) \) is submodular [3]. That is, given any two subsets \( A \subseteq B \subseteq V \) and any sensor \( v_j \in V \setminus B \), we have \( f(A \cup \{v_j\}) - f(A) \geq f(B \cup \{v_j\}) - f(B) \). It is worthy to mention that although function \( f(M) \) is submodular, \( f(M) \) is not monotonically increasing. In other words, for any two subsets \( A \subseteq B \subseteq V \), it is possible that \( f(A) > f(B) \).

We finally solve the redundant data suppression maximization problem, by reducing it to the unconstrained submodular maximization problem, and solve the problem with a \((0.5 - \epsilon)\)-approximate solution with high probability, by invoking an algorithm in [3], where \( \epsilon \) is a given constant with \( 0 < \epsilon \leq 0.5 \).

B. Finding a Randomized Solution

In the following, we briefly describe the algorithm in [3], which finds a randomized approximate solution such that the expected value of the solution is at least half of the optimal value.

Let \( V = \{v_1, v_2, \ldots, v_n\} \). The algorithm in [3] starts with two trivial solutions \( M \) and \( N \) to the problem: one solution \( M \) is that there are no master nodes, i.e., \( M = \emptyset \); and the other solution \( N \) is that all sensors are master nodes, i.e., \( N = V \). It can be seen that, in each of the two solutions, the spatial data correlations are not considered at all. The algorithm considers one sensor in \( V \) at a time in an arbitrary order, assuming that it considers sensors \( v_1, v_2, \ldots, v_n \) one by one. At the \( i \)-th iteration, the algorithm determines whether sensor \( v_i \) is a master node or a slave node with \( 1 \leq i \leq n \), by constructing two better solutions as follows.

Denote by \( M_i \) and \( N_i \) the constructed solutions after considering sensor \( v_i \), where \( 0 \leq i \leq n \). Especially, \( M_0 = \emptyset \) and \( N_0 = V \). The algorithm first calculates the marginal gain \( a_i \) by adding sensor \( v_i \) to \( M_{i-1} \), i.e., \( a_i = f(M_i \cup \{v_i\}) - f(M_{i-1}) \). It can be seen that \( a_i \) indicates the increased amount of suppressed redundant data transmissions if sensor \( v_i \) changes its role from a slave node to a master node.

The algorithm also calculates the marginal gain \( b_i = f(N_i \setminus \{v_i\}) - f(N_{i-1}) \) by removing \( v_i \) from \( N_{i-1} \), where the value of \( b_i \) indicates the increased amount of suppressed redundant data transmissions if sensor \( v_i \) changes its role from a master node in \( N_{i-1} \) to a slave node.

The algorithm then determines whether sensor \( v_i \) is a master node or a slave node by considering four cases for the values of \( a_i \) and \( b_i \): 1) \( a_i \geq 0 \) and \( b_i \geq 0 \); 2) \( a_i \geq 0 \) and \( b_i < 0 \); 3) \( a_i < 0 \) and \( b_i \geq 0 \); and 4) \( a_i < 0 \) and \( b_i < 0 \).

For case 1 that \( a_i \geq 0 \) and \( b_i \geq 0 \), this means that both changing the role of sensor \( v_i \) from a slave node in solution \( M_{i-1} \) to a master node and changing the role of \( v_i \) from a master node in \( N_{i-1} \) to a slave node will increase the amount of suppressed redundant data transmissions. The algorithm enables sensor \( v_i \) to be a master node with a probability \( \frac{a_i}{a_i + b_i} \) by adding \( v_i \) to \( M_{i-1} \), i.e., \( M_i \leftarrow M_{i-1} \cup \{v_i\} \) and \( N_i \Leftarrow N_{i-1} \); or enables \( v_i \) to be a slave node with a probability \( 1 - \frac{a_i}{a_i + b_i} = \frac{b_i}{a_i + b_i} \) by removing \( v_i \) from \( N_{i-1} \), i.e., \( M_i \Leftarrow M_{i-1} \) and \( N_i \Leftarrow N_{i-1} \setminus \{v_i\} \).

We define that \( \frac{a_i}{a_i + b_i} = 1 \) when \( a_i = 0 \) and \( b_i = 0 \). Notice that the algorithm either adds \( v_i \) to \( M_{i-1} \) or removes \( v_i \) from \( N_{i-1} \) exclusively, but not both of them at the same time.

For case 2 that \( a_i \geq 0 \) and \( b_i < 0 \), it can be seen that changing the role of sensor \( v_i \) from a slave node in solution \( M_{i-1} \) to a master node will increase the amount of suppressed redundant data transmissions \( f(M_{i-1}) \), but changing the role of \( v_i \) from a master node in \( N_{i-1} \) to a slave node will decrease the amount of suppressed redundant data transmissions \( f(N_{i-1}) \). Then, the algorithm adds \( v_i \) to \( M_{i-1} \), i.e., \( M_i \Leftarrow M_{i-1} \cup \{v_i\} \) and \( N_i \Leftarrow N_{i-1} \).

For case 3 that \( a_i < 0 \) and \( b_i \geq 0 \), it removes \( v_i \) from \( N_{i-1} \), i.e., \( M_i \Leftarrow M_{i-1} \setminus \{v_i\} \) and \( N_i \Leftarrow N_{i-1} \)

For case 4 that \( a_i < 0 \) and \( b_i < 0 \), Buchbinder et al. [3] showed that this case is impossible.
Algorithm 1 Randomized Algorithm for the redundant Data Suppression Maximization Problem

Input: n sensors v1, v2, ..., vn, the amount p_{ij} of suppressed redundant data transmissions by sensor vj if vi is its master

Output: A subset M ⊆ V of master nodes

1: Let M₀ = ∅ and N₀ = V;
2: for i ← 1 to n do
3:   Let aᵢ = f(Mᵢ₋₁ ∪ {vᵢ}) - f(Mᵢ₋₁); /* aᵢ is the marginal gain by adding vi to Mᵢ₋₁ */
4:   Let aᵢ' = max(aᵢ, 0);
5:   Let bᵢ = f(Nᵢ₋₁ \ {vᵢ}) - f(Nᵢ₋₁); /* bᵢ is the marginal gain by removing vi from Nᵢ₋₁ */
6:   Let bᵢ' = max(bᵢ, 0);
7:   if (aᵢ' + bᵢ') > 0 then
8:     Add vi to Mᵢ₋₁ with a probability aᵢ' / (aᵢ' + bᵢ'); /* Mᵢ ← Mᵢ₋₁ \ {vᵢ} and Nᵢ ← Nᵢ₋₁ \ {vᵢ} */
9:     end if
10: end for
11: Let M = Mₙ; /* in the end, Mₙ = Nₙ */
12: Each vᵢ ∈ V \ M selects a master node u ∈ M such that u = arg max_{vᵢ ∈ M} p_{ij};
13: return M.

Algorithm 2 Approximation Algorithm for the Redundant Data Suppression Maximization Problem (maxSuppression)

Input: n sensors v1, v2, ..., vn, the amount p_{ij} of suppressed redundant data transmissions by sensor vj if vi is its master node, an error ratio ϵ with 0 < ϵ ≤ 0.5, and a success probability 1 - 1/n^2

Output: A (0.5 - ϵ)-approximate solution M ⊆ V of master nodes with a probability of 1 - 1/n^2

1: Let L = [α · log_{1+2ϵ} n]; /* the number of invokings */
2: Let M ← ∅;
3: for l ← 1 to L do
4:   Find a randomized solution M(l) by invoking Algorithm 1;
5:   if f(M(l)) > f(M) then
6:     Let M ← M(l);
7:   end if
8: end for
9: return the set M of master sensors.

After n iterations, Buchbinder et al. [3] showed that Mₙ = Nₙ. Let M = Mₙ be the set of master sensors in the final solution. Buchbinder et al. [3] showed that the expected amount of suppressed redundant data transmissions is at least half of the optimal value, i.e., E[f(Mₙ)] ≥ (OPT / 2), where OPT is the optimal value of the problem. The detailed algorithm for finding a randomized solution is presented in Algorithm 1.

We illustrate the execution of Algorithm 1 with a simple example. We assume that there are three sensors v₁, v₂, and v₃ in a sensor network. Initially, M₀ = ∅ and N₀ = {v₁, v₂, v₃}. First, consider sensor v₁, assume that a₁ is much larger than b₁ and b₁ ≥ 0, this implies that a₁/(a₁ + b₁) ≈ 1. Therefore, v₁ should be added to M₀, i.e., M₁ = {v₁} and N₁ = {v₂, v₃}. Then, consider sensor v₂, assume that a₂ < 0 and b₂ > 0, which implies that v₂ should be removed from N₁, i.e., M₂ = M₁ = {v₁} and N₂ = N₁ \ {v₂} = {v₁, v₃}. Finally, consider sensor v₃, assume that a₃ < 0 and b₃ > 0, which indicates that v₃ should be removed from N₂, i.e., M₃ = M₂ = {v₁} and N₃ = N₂ \ {v₃} = {v₁}. It can be seen that M₃ = N₃ = {v₁}, sensor v₁ is the only master node while v₂ and v₃ are its slave nodes.

C. Find an Approximate Solution With High Probability

Notice that although Algorithm 1 is able to find a randomized solution by lower bounding its expected value [3] [i.e., E[f(Mₙ)] ≥ (OPT / 2)], it does not specify how much chance such a solution can be achieved. To deliver an approximate solution with high probability, we invoke Algorithm 1 multiple times, and choose the best solution among the multiple randomized solutions as the final solution to the problem. Assume that Algorithm 1 is invoked L times, and it delivers a randomized solution M(l) at the l-th time with 1 ≤ l ≤ L, where L is a given time. The final solution M to the problem then is the one with the maximum profit, i.e., f(M) = max_{l∈[1,L]} f(M(l)). The detailed algorithm for the redundant data suppression maximization problem is presented in Algorithm 2. We later show that Algorithm 2 is able to deliver a (0.5 - ϵ)-approximate solution with a probability of 1 - (1/n^2), by invoking L = [α · log_{1+2ϵ} n] times of Algorithm 1, where ϵ and α are given two constants with 0 < ϵ ≤ 0.5 and α > 0.

For example, to find a 0.45 (0.5 - 0.05) approximate solution with a probability of 1 - (1/n^2) in a network with n = 500 sensors, Algorithm 1 will be invoked L = [α · log_{1+2} n] = [1 · log_{1+2} 500] = 66 times.

D. Algorithm Analysis

Lemma 1: Given any subset M of V, define function f(M) = ∑_{vᵢ∈V\M} max_{vⱼ∈M} p_{ij}. Then, f(M) is submodular.

Proof: The proof is contained in the Appendix.

Theorem 1: Given a network G = (V, E) with n sensors v₁, v₂, ..., vₙ in V, and the amount p_{ij} of suppressed redundant data transmissions by sensor vj if vi is its master node with 1 ≤ i, j ≤ n, there is an approximation algorithm, Algorithm 2, for the redundant data suppression maximization problem, which delivers a (0.5 - ϵ)-approximate solution with a probability 1 - (1/n^2) in time O(n^2 log n log log n).

Proof: The proof is contained in the Appendix.

IV. NEAR-OPTIMAL ALGORITHM FOR THE DATA UTILITY MAXIMIZATION PROBLEM

In the previous section, we partitioned sensors into different clusters and selected a master sensor for each cluster, where the master sensor sends representative data to the sink, while
the other slave sensors in the cluster transmit only their non-redundant data. Given the clusters and chosen master sensors, in this section, we address the data utility maximization problem, by proposing a near-optimal algorithm for it.

A. Algorithm

Recall that the data utility maximization problem is to allocate the data rate $r_i(t)$ to each sensor $v_i \in V$ and the data transmission rate $f_{ij}(t)$ to each link $(v_i, v_j) \in E$ such that the accumulative utility of spatially temporally correlated data is maximized. We formally formulate the problem as

$$
\textbf{P1:} \quad \text{max} \quad \sum_{i=1}^{k} \left( U(D_i) + \sum_{v_j \in C \setminus \{u_i\}} U(D_{ij}) \right) \quad (13)
$$

subject to

$$
\sum_{v_j \in N(v_i)} f_{ij}(t) + r_i(t) = \sum_{v_j \in N(v_i)} f_{ij}(t) \quad \forall v_i \in V, \quad 1 \leq t \leq T \quad (14)
$$

where constraint (14) indicates that sensor $v_i$ will forward its received data and its sensing data to the sink via the relay of its neighbors; constraint (15) describes the relationship between the residual energy $RE(v_i, t)$ and $RE(v_i, t - 1)$ of each sensor $v_i$ at time slot $t$ and its previous time slot $t - 1$; $B_i$ is the battery capacity of $v_i$; $0 \leq RE(v_i, t) \leq B_i$; $f_{ij}(t) \geq 0$; and $0 \leq r_i(t) \leq R_i$.

Notice that problem $\textbf{P1}$ is not a convex optimization problem, as both its objective function and constraint (15) are not convex. Fortunately, we can transform it into a convex optimization problem $\textbf{P2}$

$$
\textbf{P2:} \quad \text{min} \quad \sum_{i=1}^{k} \left( U(D_i) + \sum_{v_j \in C \setminus \{u_i\}} U(D_{ij}) \right) \quad (16)
$$

subject to constraint (14) and

$$
RE(v_i, t) \leq RE(v_i, t - 1) + H(v_i, t) - P(v_i, t). \quad (17)
$$

Following a recent study [37], the optimal values of problem $\textbf{P1}$ and $\textbf{P2}$ are equal. Then, we can find a near-optimal solution to the convex optimization problem $\textbf{P2}$, by applying any existing algorithm, such as interior-point methods, subgradient methods, ellipsoid methods, cutting-planes methods, etc. [1]. The solution to $\textbf{P2}$ in turn returns a near-optimal solution to $\textbf{P1}$. The detailed algorithm for problem $\textbf{P1}$ is termed as algorithm maxUtility.

Theorem 2: Given a sensor network $G = (V \cup \{s\}, E)$, a data collection period $T$, the residual energy $RE(v_i, 0)$ of each sensor $v_i$ at the beginning of period $T$, the harvested energy $H(v_i, t)$ of each sensor $v_i$ at time slot $t$ with $1 \leq t \leq T$, there is a near-optimal algorithm for the data utility maximization problem in $G$.

Proof: Following a recent study in [37], the optimal values of problem $\textbf{P1}$ and $\textbf{P2}$ are equal. Following the work [1], the barrier method in the existing algorithm can find a solution with its total utility only $\epsilon$ smaller than the maximum utility, where $\epsilon$ is a given additive error with $0 < \epsilon < 1$. The theorem then follows.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed algorithms, by comparing them with the state-of-the-art algorithms for the clustering problem and the data utility maximization problem. We also investigate the impact of different parameters on the performance.

A. Simulation Settings

We consider a sensor network, which consists of from 50 to 200 sensors. The sensors are deployed in a $1000 \times 1000$ square area randomly. The sink $s$ is located at the center of the area. Assume that the battery capacity of each sensor is $B = 10.8$ $kJ$ and the maximum sensing rate of the sensor is $R = 10$ $kb/s$ [37]. We consider a monitoring period $T = 24$ hours that is divided into equal time slots with an equal duration of $\tau$ (e.g., $\tau = 1$ hour). Each sensor is powered by a 37 mm $\times$ 33 mm solar panel. We adopt real 89 solar power harvesting profiles from the baseline measurement system at the National Renewable Energy Laboratory [25].

Following a recent study in [37], the optimal values of problem $\textbf{P1}$ and $\textbf{P2}$ are equal. Then, we can find a near-optimal solution to the convex optimization problem $\textbf{P2}$, by applying any existing algorithm, such as interior-point methods, subgradient methods, ellipsoid methods, cutting-planes methods, etc. [1]. The solution to $\textbf{P2}$ in turn returns a near-optimal solution to $\textbf{P1}$. The detailed algorithm for problem $\textbf{P1}$ is termed as algorithm maxUtility.

Theorem 2: Given a sensor network $G = (V \cup \{s\}, E)$, a data collection period $T$, the residual energy $RE(v_i, 0)$ of each sensor $v_i$ at the beginning of period $T$, the harvested energy $H(v_i, t)$ of each sensor $v_i$ at time slot $t$ with $1 \leq t \leq T$, there is a near-optimal algorithm for the data utility maximization problem in $G$.

We adopt both real and simulation sensing data in our experiments. On the one hand, we use real temperature sensing data collected from 54 sensors for 38 days (from February 28 to April 5 in 2004) [12]. In our experiments, the sensing data sequence of a sensor $v_i$ on day $d$ is randomly chosen from the 54 sensor data sequence on a random day of the 38 days. Then, the spatial data correlations of nearby sensors follow the distribution in Fig. 4. On the other hand, we use simulation data as follows. For any two neighboring sensors $v_i$ and $v_j$, the value of their spatial data correlation $c(i,j)$ is randomly chosen from an interval $[0, C_{max}]$ with $C_{max} = 1$.

To evaluate the performance of the proposed algorithms maxSuppression and maxUtility, we adopt five existing algorithms.

1. Algorithm maxThroughput [24] does not take temporal and spatial data correlations, and it aims to maximize the total amount of data collected in a period.

2. Algorithm maxTemporalUtility [37] considered only temporal data correlation and focused the problem
of sensor data rates and routing allocations, such that the accumulative utility of temporally correlated data is maximized.

3) Algorithm YEAST [32] exploited only spatial data correlation, by partitioning sensors into different clusters and selecting representative nodes to construct a dynamic scalable routing tree to minimize sensor energy consumption.

4) Algorithm maxQuality [16], [23] incorporated both temporal and spatial data correlations, through fair rate allocations and flow routing to maximize the quality of collected data in an energy harvesting sensor network.

5) Algorithm 2TC-cor [31] also incorporated both temporal and spatial data correlations and proposed a subclustering algorithm based on spatial data correlation, in which each sensor first compares its generated data with its historical data, so as to suppress transmissions of similar temporal data, cluster heads then select some representative sensors for further energy saving by considering spatial correlation.

B. Algorithm Performance

We first study the performance of the different algorithms by increasing the network size $n$ from 50 to 200. Fig. 6(a) shows that the amount of suppressed redundant data transmissions by the proposed algorithm maxSuppression is from 50% to 55% larger than those by algorithms maxQuality, YEAST, and 2TC-cor, when $n$ increases from 50 to 200. Notice that we here did not plot the performance of algorithms maxThroughput and maxTemporalUtility, as they did not take the spatial data correlations among sensors into consideration, and thus the amounts of suppressed redundant data transmissions by them are zeros. Fig. 6(b) demonstrates that the accumulative utility of the collected data by the proposed algorithm maxUtility is about 17% larger than those by the other algorithms. For example, the accumulative utilities by maxUtility, maxQuality, YEAST, 2TC-cor, maxTemporalUtility, and maxThroughput are 690, 590, 533, 522, 480, and 454, respectively, when there are 200 sensors. Fig. 6(c) further plots the average throughput per sensor by the different algorithms, from which it can be seen that the average throughput per sensor by algorithm maxThroughput decreases dramatically with the distance to the sink. In contrast, the average throughput per sensor by the proposed algorithm maxUtility is much larger than those by the other five algorithms when the distance to the sink is longer than 200 m. For example, the average throughputs by maxUtility, maxQuality, YEAST, 2TC-cor, maxTemporalUtility, and maxThroughput are 41, 23, 16, 17, 12, and 10 Mb, respectively, when the distance to the sink is 400 m. The rationale behind is that the proposed algorithm maxUtility can exploit much more spatial data correlations, thereby collecting fewer amounts of data from nearby sensors and saving their energy for relaying the nonredundant data from remote sensors.

We then investigate the algorithm performance by increasing the maximum sensing rate $R$ from 1 to 10 kb/s. Fig. 7(a) shows that the difference between the amount of suppressed redundant data transmissions by the proposed algorithm maxSuppression and the amounts by other algorithms becomes larger with the increase of the maximum sensing rate $R$, as there are more redundant data if each sensor generates data faster. Fig. 7(a) also demonstrates that the amount of suppressed redundant data transmissions by algorithm maxSuppression is about 50% larger than those by other existing algorithms. Fig. 7(b) shows that the data utility by the algorithm maxUtility is from 19% to 29% larger than those by other algorithms, and the utility by each algorithm increases with the growth of $R$ since more redundant information will be sent. Fig. 7(c) demonstrates that the throughputs of different sensors by algorithm maxUtility distributes more fairly than those by other algorithms. Notice that the average throughput per sensor by the proposed algorithm maxUtility is smaller than those by the existing algorithms when the distance to the sink is small. The rationale behind is as follows. This article addresses the problem of collecting nonredundant data from sensors, by incorporating both spatial data correlations and temporal data correlations. The solutions delivered by the existing algorithms will collect a large amount of data from the sensors near to the sink, but less amount of data from remote sensors. However, the collected data from the nearby sensors are highly redundant. In contrast, the solution delivered by the proposed algorithm maxUtility collects more data from remote sensors, though the amount of data collected from nearby sensors is less than those by the existing studies. The utility of the data collected,
i.e., the amount of nonredundant data, by the proposed algorithm \( \text{maxUtility} \) is larger than those by the existing algorithm.

We finally study the algorithm performance by increasing the maximum spatial data correlation \( C_{\text{max}} \) from 0 to 1. Fig. 8(a) shows that the amount of suppressed redundant data transmissions by the proposed algorithm \( \text{maxSuppression} \) is about from 50% to 60% larger than those by the other algorithms when \( C_{\text{max}} \) increases from 0.2 to 1. However, the amounts of suppressed redundant data transmissions by each algorithm is 0 when \( C_{\text{max}} = 0 \) since \( C_{\text{max}} = 0 \) indicates that there are no spatial data correlations. Fig. 8(b) shows that the accumulative data utility by each of the four algorithms \( \text{maxUtility} \), \( \text{maxQuality} \), YEAST, and 2TC-cor increase with the growth of \( C_{\text{max}} \), while the utilities by both \( \text{maxTemporalUtility} \) and \( \text{maxThroughput} \) do not change with the increase of \( C_{\text{max}} \), as both they do not consider spatial data correlations among sensors. In addition, Fig. 8(b) plots that the difference between the data utility by algorithm \( \text{maxUtility} \) and those by the other algorithms increase with the growth of \( C_{\text{max}} \). Especially, the utility by algorithm \( \text{maxUtility} \) is 20% larger than those by the other algorithm when \( C_{\text{max}} = 1 \). Fig. 8(c) finally demonstrates that the average throughput per sensor by the proposed algorithm \( \text{maxUtility} \) is much larger than those by the other five algorithms when the distances to the sink are longer than 200 m, but less amount of data from the sensors with their distance to the sink being less than 100 m. This implies that the solution delivered by the proposed algorithm \( \text{maxUtility} \) collects data from different sensors more fairly, rather than from only a few nearby sensors. Therefore, the utility by the proposed algorithm is larger than those by the existing algorithms.

VI. RELATED WORK

There are extensive studies on efficient data collection in WSNs in the past. Some existing studies mainly focused on optimizing sensor energy saving when performing sensing data collection, without the consideration of temporal and spatial data correlations [5], [13], [15], [24], [26], [34]. For example, Liew et al. [18] proposed a fast, adaptive, and energy-efficient data collection protocol for only battery-powered sensor networks, by utilizing multichannel and multipath data transmission, thereby reducing delay and packet loss rates. He et al. [10] proposed a data gathering scheme with a mobile sink to achieve efficient data collection in sensor networks. Liu et al. [21] designed a distributed routing algorithm for data collection in low duty cycle WSNs, which achieves a better tradeoff between data transmission latency and sensor energy conservation than existing schemes. Also, there are some studies focusing on replenishing sensor energy through harvesting energy from their surrounding environments [24], [26]. Gallego et al. [7] presented a medium access control protocol that combines distributed queuing and energy harvesting. Mao et al. [24] studied the problem of allocating energy for data sensing and data transmitting, and used the infinite horizon Markov decision process to maximize the amount of collected data. Mehrabi and Kim [26] and Wang et al. [33] studied the use of mobile sinks to collect data from energy-harvested sensors such that the amount of...
data collected is maximized. Xu et al. [34] investigated the employment of multiple energy-constrained sinks to collect sensor data to maximize the amount of data collected, subject to the constraint on the energy capacity of each sink.

Although extensive studies have been conducted for data collections and energy conservation in sensor networks [24], [26], they did not take into account sensing data correlations. Deng et al. [6] and Zhang et al. [37] recognized that sensing data are quite similar during a short period of time, that is, sensing readings are temporally correlated. Deng et al. [6] studied the network utility maximization problem in static routing rechargeable sensor networks and allocated sensing rates for sensors. They devised a distributed algorithm to obtain globally optimal data rates. Zhang et al. [37] devised a near-optimal algorithm for the utility maximization problem, by considering time-varying energy harvesting rates of sensors.

We also noticed that some researchers found that sensing data are spatially correlated and they aimed to maximize data quality by utilizing spatial correlation while reducing energy consumption [22], [32], [35]. For example, Yoon and Shahabi [35] designed an algorithm to partition sensors into multiple clusters such that sensors in each cluster are close with each other and their sensing data are highly similar. Then, only the cluster heads transmit their data to the sink, whereas the other sensors do not generate data, thereby saving their energy. Villas et al. [32] focused on the cluster head choice by considering the residual energy of sensors and designed an efficient routing algorithm for data collection. Liu et al. [22] strived for the fine balance between the quality of collected data and sensor energy consumption, by choosing a sensor in each cluster to work with a given probability \( p \). That is, a smaller value of \( p \) is chosen if the residual energy of a sensor is low, while a larger \( p \) is adopted when the monitoring quality needs to be improved. Brahmi et al. [2] devised a hybrid strategy for energy-efficient data gathering: local aggregation and global aggregation, where the local aggregation means that data from member sensors in the same cluster are aggregated by the cluster head, while the global aggregation indicates a cluster head also aggregates the data received from cluster heads. In order to further improve data collection and energy efficiency, some studies considered both energy harvesting and spatial correlations. Qamaji and Atakan [27] studied the tradeoff between energy conservation and data distortion level.

There are quite a few studies that considered both the temporal and spatial data correlations [16], [30], [31], [36]. For example, Liang et al. [16] first clustered sensors by their spatial data correlation such that each cluster has no more than two sensors, one sensor serves as the master sensor while the other serves the slave sensor. They then investigated a fair data rate allocation problem. Tayeh et al. [30] assumed that a sufficient number of powerful cluster heads have been deployed so that each sensor can communicate with at least one cluster head, and then each cluster head adjusts the data rates of the sensors associated with it. However, it is very challenging to select cluster heads in a sensor network, which must consider both the spatial data correlations among sensors and their residual energy. Tsai and Huang [31] proposed a subclustering algorithm based on spatial data correlation, in which each member sensor first compares its generated data with its historical data, so as to suppress the transmissions of similar data, cluster heads then select some representative sensors for further energy saving by considering spatial correlation. Yoon and Shahabi [36] extended their work in [35], respectively, by incorporating temporal data correlation, whereas the latter study in [35] considered only spatial correlation. Liang et al. [17] studied a problem of scheduling a UAV to collect spatially correlated data from PoIs such that the amount of nonredundant information collected is maximized, subject to the energy capacity constraint on the UAV.

It can be seen that although the existing studies considered both spatial and temporal data correlation, they adopted a coarse-grained spatial correlation model. That is, each sensor either transmits its data to the sink or not. Unlike the aforementioned studies, we here proposed a fine-grained spatial correlation model, in which some representative sensors send all their sensing data, while the others transmit only nonredundant data from that sent by the representative sensors. Therefore, our work is promising in reducing the transmissions of redundant data, while maintaining a high quality of collected data.

VII. CONCLUSION

Unlike the existing studies that adopted the coarse-grained spatial-correlation model, in this article, we introduced a novel fine-grained spatial-correlation model, in which only a small portion of sensors send their representative data and the other majority sensors transmit their nonredundant data, thereby reducing redundant data transmissions. We first studied the redundant data suppression maximization problem under the proposed model, by proposing a novel approximation algorithm for it, which delivers a \((0.5 - \epsilon)\)-approximate solution with high probability, where \( \epsilon \) is a given constant with \( 0 < \epsilon \leq 0.5 \). We then devised a near-optimal algorithm for the data utility maximization problem such that accumulative utility of the spatially temporally correlated data received by the sink is maximized. We finally evaluated the performance of the proposed algorithms, using real data sets. The experimental results show that the proposed algorithms outperform the existing algorithms.

APPENDIX

A. Algorithm Analysis

We show that function \( f(M) \) is submodular by the following lemma.

**Lemma 2:** Given any subset \( M \) of \( V \), define function \( f(M) = \sum_{v_j \in V \setminus M} \max_{v_i \in M} \{ p_{ij} \} \). Then, \( f(M) \) is submodular.

**Proof:** Given any two subsets \( A \subseteq B \subseteq V \) and any sensor \( v_k \in V \setminus B \), we show that \( f(A \cup \{ v_k \}) - f(A) \geq f(B \cup \{ v_k \}) - f(B) \) as follows.

For any \( v_j \in V \setminus (B \cup \{ v_k \}) \), it can be seen that

\[
\max_{v_i \in A \cup \{ v_k \}} \{ p_{ij} \} - \max_{v_i \in A} \{ p_{ij} \} = \max \left\{ p_{kj} - \max_{v_i \in A} \{ p_{ij} \}, 0 \right\}
\]
\[ \geq \max \left\{ p_{ij} - \max_{v_i \in A} \{ p_{[j]} \} , 0 \right\} \]

as \( \max_{v_i \in A} \{ p_{[j]} \} \leq \max_{v_i \in B} \{ p_{[j]} \} \), due to \( A \subseteq B \)

\[ = \max_{v_i \in B \cup \{ v_k \}} \{ p_{[j]} \} - \max_{v_i \in B} \{ p_{[j]} \} \]

We then have

\[ f(A \cup \{ v_k \}) - f(A) \]
\[ = \sum_{v_j \in (A \cup \{ v_k \})} \max_{v_i \in A \cup \{ v_k \}} \{ p_{[j]} \} - \sum_{v_j \in A} \max_{v_i \in A} \{ p_{[j]} \} \]
\[ = \sum_{v_j \in (A \cup \{ v_k \})} \max_{v_i \in A \cup \{ v_k \}} \{ p_{[j]} \} - \sum_{v_j \in A} \max_{v_i \in A} \{ p_{[j]} \} \]
\[ - \max_{v_i \in A} \{ p_{[k]} \} \]
\[ = \sum_{v_j \in (A \cup \{ v_k \})} \left( \max_{v_i \in A \cup \{ v_k \}} \{ p_{[j]} \} - \max_{v_i \in A} \{ p_{[j]} \} \right) - \max_{v_i \in A} \{ p_{[k]} \} \]
\[ \geq \sum_{v_j \in (B \cup \{ v_k \})} \left( \max_{v_i \in B \cup \{ v_k \}} \{ p_{[j]} \} - \max_{v_i \in B} \{ p_{[j]} \} \right) - \max_{v_i \in A} \{ p_{[k]} \} \]

as \( V \setminus (A \cup \{ v_k \}) \supseteq V \setminus (B \cup \{ v_k \}) \), due to \( A \subseteq B \)

\[ \geq \sum_{v_j \in (B \cup \{ v_k \})} \left( \max_{v_i \in B \cup \{ v_k \}} \{ p_{[j]} \} - \max_{v_i \in B} \{ p_{[j]} \} \right) - \max_{v_i \in A} \{ p_{[k]} \} \]
due to (18)

\[ \geq \sum_{v_j \in B \cup \{ v_k \}} \left( \max_{v_i \in B \cup \{ v_k \}} \{ p_{[j]} \} - \max_{v_i \in B} \{ p_{[j]} \} \right) - \max_{v_i \in A} \{ p_{[k]} \} \]
since \( \max_{v_i \in A} \{ p_{[k]} \} \leq \max_{v_i \in B} \{ p_{[k]} \} \), due to \( A \subseteq B \)

\[ = \sum_{v_j \in B \cup \{ v_k \}} \max_{v_i \in B \cup \{ v_k \}} \{ p_{[j]} \} - \sum_{v_j \in B \cup \{ v_k \}} \max_{v_i \in B} \{ p_{[j]} \} \]
\[ - \max_{v_i \in B} \{ p_{[k]} \} \]
\[ = \sum_{v_j \in B \cup \{ v_k \}} \max_{v_i \in B \cup \{ v_k \}} \{ p_{[j]} \} - \sum_{v_j \in B \cup \{ v_k \}} \max_{v_i \in B} \{ p_{[j]} \} \]
\[ = f(B \cup \{ v_k \}) - f(B) \]
\[ \geq (1-\epsilon) \cdot \text{OPT} \]
\[ \leq \text{OPT} \]

Algorithm 2 is \( M = \text{arg max}_{i=1} \{ f(M(i)) \} \). In the following, we first analyze the probability that \( f(M) \geq (0.5 - \epsilon) \cdot \text{OPT} \).

Let \( \pi_i \) be the probability that the value of the \( M(i) \) is no more than \((0.5 - \epsilon) \cdot \text{OPT}\), i.e.,

\[ \pi_i = P_r \left[ f(M(i)) \leq (0.5 - \epsilon) \cdot \text{OPT} \right] \]

where \( 1 \leq i \leq L \). The expected value of \( M(i) \) then can be upper bounded by

\[ \mathbb{E} \left[ f(M(i)) \right] \leq \pi_i \cdot (0.5 - \epsilon) \cdot \text{OPT} + (1 - \pi_i) \cdot \text{OPT} \]

Following Buchbinder et al. [3], we have \( \mathbb{E} \left[ f(M(i)) \right] \geq (\text{OPT}/2) \). Then, the value of probability \( \pi_i \) is thus is upper bounded by

\[ \pi_i \leq \frac{1}{1 + 2\epsilon} \]

The probability that the value of each of the \( L \) randomized solutions is no more than \((0.5 - \epsilon) \cdot \text{OPT}\) then is no more than

\[ \prod_{i=1}^{L} \pi_i \leq \left( \frac{1}{1 + 2\epsilon} \right)^L \]
\[ = \left( \frac{1}{1 + 2\epsilon} \right)^{\frac{\alpha \cdot \log_2 n}{\epsilon}} \]
\[ \leq \left( \frac{1}{1 + 2\epsilon} \right)^{\alpha \cdot \log_2 n} \]
\[ = \frac{1}{n^\alpha} \]

This indicates that the probability that Algorithm 2 delivers a \((0.5 - \epsilon)\)-approximate solution is no less than \(1 - (1/n^\alpha)\).

The rest is to analyze the time complexity of Algorithm 2.

In the following, we show that Algorithm 1 can be efficiently implemented in time \( O(m \log \Delta) \). Then, the time complexity of Algorithm 2 is \( O(Lm \log \Delta) = O(\alpha \cdot (m \log \Delta) \log_2 n) \).

Notice that the running time of Algorithm 1 is dominated by calculating \( a_i \) and \( b_i \) with \( 1 \leq i \leq n \), which is analyzed as follows.

We first analyze the time complexity of the calculation of \( a_i \) for \( 1 \leq i \leq n \). Before each iteration \( i \), assume that we already knew the value \( \max_{v_i \in M(i-1)} \{ p_{[k]} \} \) for each \( v_i \in V \setminus M(i-1) \). Notice that the addition of \( v_i \) to \( M(i-1) \) will change the values \( \max_{v_i \in M(i-1)} \{ p_{[k]} \} \) for only neighbors \( v_j \) of \( v_i \), but not change the value \( \max_{v_j \in M(i-1)} \{ p_{[k]} \} \) for a sensor \( v_j \) such that \( v_j \notin N(v_i) \), since \( \max_{v_j \in M(i-1)} \{ p_{[k]} \} = \max_{v_j \in N(v_i) \setminus M(i-1)} \{ p_{[k]} \} = \max_{v_j \in N(v_i) \setminus M(i-1)} \{ p_{[k]} \} = \max_{v_j \in M(i-1)} \{ p_{[k]} \} \). Due to that \( v_i \) is not a neighbor of \( v_j \). Then

\[ a_i = f(M(i-1) \cup \{ v_i \}) - f(M(i-1)) \]
\[ = \sum_{v_j \in N(v_i) \setminus M(i-1)} \left( \max_{v_k \in M(i-1) \cup \{ v_i \}} \{ p_{[j]} \} - \max_{v_k \in M(i-1)} \{ p_{[j]} \} \right) \]
\[ - \max_{v_k \in M(i-1)} \{ p_{[k]} \} \]
\[ = \max_{v_i \in M(i-1)} \{ p_{[j]} \} - \max_{v_i \in M(i-1)} \{ p_{[j]} \} \]
\[ - \max_{v_i \in M(i-1)} \{ p_{[k]} \} \]
\[ \geq \max_{v_i \in M(i-1)} \{ p_{[j]} \} - \max_{v_i \in M(i-1)} \{ p_{[j]} \} \]
\[ - \max_{v_i \in M(i-1)} \{ p_{[k]} \} \]
Since the value $\max_{v_j \in N_{i-1}} |p_{kj}|$ for each $v_j \in V \setminus M_{i-1}$ is already known, the calculation of $a_i$ takes time $O(|N(v_i)|) = O(\delta_i)$, where $\delta_i = |N(v_i)|$. Then, the calculation for all $a_i$s takes time $\sum_{i=1}^m O(\delta_i) = O(m)$, where $m$ is the number of edges in $G$, i.e., $m = |E|$

We then analyze the time complexity of the calculations of all $b_i$s. Similarly, we assume that we already know the value $\max_{v_j \in N_{i-1}} |p_{kj}|$ for each $v_j \in V \setminus N_{i-1}$ before each iteration $i$. Also, if a sensor $v_i$ is removed from $N_{i-1}$ at the $i$th iteration in Algorithm 1, we construct a list $L_i$ of neighbors of $v_i$, where a sensor $v_k$ is in $L_i$ if and only if $v_k \in N(v_i)$ and $v_k \in N_i \setminus \{v_i\}$. We sort the neighbors in list $L_i$ in decreasing order of their values $p_{ki}$. Then, the calculation of the lists in the $n$ iterations of Algorithm 1 takes time $\sum_{i=1}^n O(\delta_i) \log \delta_i = \sum_{i=1}^n O(\delta_i) \log \Delta = O(m \log \Delta)$, where $\delta_i = |N(v_i)|$ and $\Delta = \max_{v_j \in N_{i-1}} |p_{kj}|$

Notice that the removal of $v_i$ from $N_{i-1}$ will change the value $\max_{v_j \in N_{i-1}} |p_{kj}|$ for only neighbors of $v_i$, but do not change the value $\max_{v_j \in N_{i-1}} |p_{kj}|$ for a sensor $v_j$ such that $v_j \notin N(v_i)$, since $\max_{v_j \in N_{i-1}} |p_{kj}| = \max_{v_j \in N(v_i) \cap N_{i-1}} |p_{kj}| = \max_{v_j \in N_{i-1}} |p_{kj}|$, due to the fact that $v_i$ is not a neighbor of $v_j$. Then

$$b_i = f(N_{i-1} \setminus \{v_i\}) − f(N_{i-1}) = \sum_{v_j \in N(v_i) \cap N_{i-1}} \left( \max_{v_k \in N_{i-1} \setminus \{v_i\}} |p_{kj}| − \max_{v_k \in N_{i-1}} |p_{kj}| \right) + \max_{v_k \in N_{i-1}} |p_{ki}|$$

To calculate the value of $\theta_j$, we search the neighbor list $L_j$ of $v_j$ from the beginning to the end since the neighbors of $v_j$ in $L_j$ are sorted in decreasing order of $p_{ki}$. Denote by $L_j = v_j, 1, v_j, 2, \ldots, v_j, n_j$, where $n_j$ is the number of nodes in $L_j$. If the first node $v_j, 1$ is in set $N_{i-1} \setminus \{v_i\}$, then $\max_{v_k \in N(v_i) \cap N_{i-1}} |p_{kj}| = \max_{v_k \in N_{i-1}} |p_{kj}| = p_{v_i, v_j, 1}$, and $\theta_j = 0$. Otherwise ($v_j, 1$ is not in $N_{i-1} \setminus \{v_i\}$), we remove $v_j, 1$ from $L_j$ and search the next node, until we find a node $v_j, k$ such that $v_j, k$ is in $N_{i-1} \setminus \{v_i\}$, where $1 \leq k \leq n_j$. Then, $\max_{v_k \in N(v_i) \cap N_{i-1}} |p_{kj}| = p_{v_i, v_j, k}$. Denote by $n_{ij}$ the number of nodes removed from $L_j$ before we find the node $v_j, k$. Then, the calculation of $b_i$ takes time

$$\sum_{v_j \in N(v_i) \cap N_{i-1}} (O(n_{ij}) + O(1)) + O(\delta_i) \leq \sum_{v_j \in N(v_i) \cap N_{i-1}} (O(n_{ij}) + O(1)) + O(\delta_i) = \sum_{v_j \in N(v_i) \cap N_{i-1}} O(n_{ij}) + O(\delta_i)$$

where $\delta_i = |N(v_i)|$ and let $n_{ij} = 0$ for $v_j \notin N(v_i) \setminus N_{i-1}$.

The calculation of all $b_i$s takes time

$$O(m \log \Delta) + \sum_{i=1}^n \left( \sum_{v_j \in N(v_i) \cap N_{i-1}} O(n_{ij}) + O(\delta_i) \right) = O(m \log \Delta) + \sum_{i=1}^n \sum_{v_j \in N(v_i) \cap N_{i-1}} O(n_{ij}) + \sum_{i=1}^n O(\delta_i) = O(m \log \Delta) + \sum_{i=1}^n \sum_{v_j \in N(v_i) \cap N_{i-1}} O(n_{ij}) + O(m) \leq O(m \log \Delta) + \sum_{i=1}^n O(\delta_i) + O(m) = O(m \log \Delta) + O(m) + O(m) = O(m \log \Delta)$$

where (27) holds since each node in list $L_i$ will be removed no more than once, which indicates that the number of removed nodes from $L_i$ is no more than the number of nodes in $L_i$ when $L_i$ is constructed, i.e., $\sum_{v_j \in N(v_i) \cap N_{i-1}} O(n_{ij}) \leq O(\delta_i)$

Therefore, the time complexity of Algorithm 1 is $O(m) + O(m \log \Delta) = O(m \log \Delta)$. Then, the time complexity of Algorithm 2 is $O(\alpha m (\log \Delta) \log_2 (\alpha n) \log n) = O(m (\log n)^2)$, as $\Delta \leq n$ and $\alpha$ and $\epsilon$ are given two constants. The theorem then follows.

**ACKNOWLEDGEMENT**

The authors would like to thank the anonymous referees and the associate editor for their constructive comments and valuable suggestions, which helped to improve the quality and presentation of the article greatly.

**REFERENCES**


Weigang Wu (Member, IEEE) received the B.Sc. and M.Sc. degrees from Xi’an Jiaotong University, Xi’an, China, in 1998 and 2003, respectively, and the Ph.D. degree in computer science from Hong Kong Polytechnic University, Hong Kong, in 2007.

He is currently a Full Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. He has published more than 60 papers in major conferences and journals. His research interests include distributed systems and wireless networks, especially, cloud computing platforms, and ad hoc networks.

Prof. Wu has served as a member of the editorial board of two international journals, Frontiers of Computer Science and Ad Hoc and Sensor Wireless Networks. He is also an Organizing/Program Committee Member for many international conferences. He is a member of ACM.

Bing Guo received the B.S. degree in computer science from Beijing Institute of Technology, Beijing, China, in 1991, and the M.S. and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 1999 and 2002, respectively.

He is currently a Professor and Vice Dean with the School of Computer Science, Sichuan University, Chengdu. His current research interests include embedded real-time system and green computing.

Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees in computer science from Dalian University of Technology, Dalian, China, in 2011 and 2008, respectively, and the Ph.D. degree from Australian National University, Canberra, ACT, Australia, in 2016.

He was a Research Associate with University College London, London, U.K. He is currently an Associate Professor with the School of Software, Dalian University of Technology. His research interests include cloud computing, software-defined networking, wireless sensor networks, algorithmic game theory, and optimization problems.

Yue Ivan Wu (Member, IEEE) received the B.Eng. degree in communication engineering and the M.Eng. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 2004 and 2007, respectively, and the Ph.D. degree in electronic and information engineering from Hong Kong Polytechnic University, in 2010.

He was a Postdoctoral Fellow or a Research Fellow in 2011 and from 2015 to 2016 with Hong Kong Polytechnic University. He was a Lecturer with the University of Electronic Science and Technology of China from May 2010 to June 2013. He joined VALENS with Nanyang Technological University, Singapore, as a Research Fellow in June 2011, and then the University of Florida, Gainesville, FL, USA, as a Postdoctoral Associate in July 2012. Since July 2013, he has been an Associate Professor with the College of Computer Science, Sichuan University, Chengdu. He currently works as a Project Director with the Department of Information Science, National Natural Science Foundation of China. His research interests include space-time signal processing, wireless sensor networks, and wireless channel modeling.

Dr. Wu is on the editorial boards of IET Signal Processing, IEEE ACCESS, and Telecommunication Systems.