

Research Article

Minimum Cost Data Aggregation for Wireless Sensor Networks Computing Functions of Sensed Data

Chao Chen,¹ Kyogu Lee,² Joon-Sang Park,³ and Seung Jun Baek¹

¹Department of Computer and Communications, Korea University, Seoul 136-701, Republic of Korea

²Department of Digital Contents Convergence, Seoul National University, Seoul 151-742, Republic of Korea

³Department of Computer Engineering, Hongik University, Seoul 121-791, Republic of Korea

Correspondence should be addressed to Seung Jun Baek; sjbaek@korea.ac.kr

Received 11 December 2014; Accepted 12 January 2015

Academic Editor: Yun Liu

Copyright © 2015 Chao Chen et al. 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.

We consider a problem of minimum cost (energy) data aggregation in wireless sensor networks computing certain functions of sensed data. We use in-network aggregation such that data can be combined at the intermediate nodes en route to the sink. We consider two types of functions: firstly the summation-type which includes *sum*, *mean*, and *weighted sum*, and secondly the extreme-type which includes *max* and *min*. However for both types of functions the problem turns out to be NP-hard. We first show that, for *sum* and *mean*, there exist algorithms which can approximate the optimal cost by a factor logarithmic in the number of sources. For *weighted sum* we obtain a similar result for Gaussian sources. Next we reveal that the problem for extreme-type functions is intrinsically different from that for summation-type functions. We then propose a novel algorithm based on the crucial tradeoff in reducing costs between local aggregation of flows and finding a low cost path to the sink: the algorithm is shown to empirically find the best tradeoff point. We argue that the algorithm is applicable to many other similar types of problems. Simulation results show that significant cost savings can be achieved by the proposed algorithm.

1. Introduction

Motivation. In this paper we consider the problem of minimum cost (energy) data aggregation in wireless sensor networks (WSN) where the aggregated data is to be reported to a single sink. A common objective of WSN is to retrieve certain *summary* of sensed data instead of the entire set of data. The relevant summary is defined as a certain function applied to a set of measured data [1]. Specifically we are given a function $g(\cdot)$ such that, for a set of measurement data x_1, \dots, x_n , the goal of the sink is to retrieve $g(x_1, x_2, \dots, x_n)$. Examples of $g(\cdot)$ are *mean*, *max*, *min*, and so forth. When *mean* function is used, $g(x_1, \dots, x_n) = \sum_{i=1}^n x_i/n$. For applications such as “alarm” systems, one can use *max* as $g(\cdot)$, for example, $g(x_1, \dots, x_n) = \max_{i=1, \dots, n} [x_i]$ where x_i can be temperature values in forest-fire monitoring systems or the structural stress values measured in a building. We will refer to $g(\cdot)$ as a *summary function* throughout this paper. Certain types of $g(\cdot)$ allow us to combine data at the intermediate

nodes en route to the sink. Such combining techniques are commonly referred to as *in-network aggregation* [2–4]. By using in-network aggregation one can potentially save communication costs by reducing the amount of traffic [5–7]. For instance, in the applications such as wireless multimedia sensor networks (WMSN) where the transmitted multimedia data has a far greater volume than that in typical WSNs, the in-network aggregation technique is crucial for the purpose of saving energy and extending network lifetime [8, 9]. While in-network aggregation offers many benefits, it poses significant challenge for network design, for example, designing routing algorithms so as to minimize costs such as energy expenditure and delay. In particular, we show that it is crucial to take into account how the summary function $g(\cdot)$ affects the statistical properties of sensed data.

Objectives. In this paper we study the minimum cost aggregation problem for several types of $g(\cdot)$. The performance of in-network aggregation relies heavily on the properties of

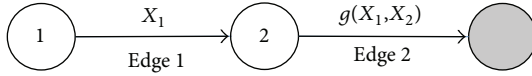


FIGURE 1: An example of computing and communicating a summary.

the function $g(\cdot)$. To be specific let us briefly look at the problem formulation. Consider the single-sink aggregation problem where we define the cost function as follows. Let E denote the set of links in the network. We would like to minimize

$$\sum_{e \in E} w_e \phi_e, \quad (1)$$

where w_e represents the weight associated with link e and ϕ_e represents the average number of bits transmitted over e . Note that the objective similar to (1) has been considered in [10–14] as well. The most relevant objective associated with (1) is the *energy consumption*. To see this, let us define weight $w_e := k_e d_e^{-\alpha}$ where d_e is the distance between nodes connected by Link e , α is the path loss exponent, and k_e is the related channel parameter. Hence (1) is proportional to the total transmitted energy consumed throughout the data aggregation. Note in [13, 14], the authors consider the same energy cost function. We refer to ϕ_e as the *aggregation cost function* (we will use notation ϕ to denote the cost function in general, whereas ϕ_e is used to denote the cost function specifically on Link e). Note that ϕ_e depends on the source measurements aggregated on e , and also on $g(\cdot)$ which is the summary function applied to the measurements. The work in [15] also studies an aggregation problem in sensor networks computing summary functions, assuming that all the packets generated in the network have the same size. However, the amount of information generated at intermediate nodes may vary, since a summary of data can be statistically different from the original data, which is our key observation.

Let us take an example. Consider the network in Figure 1 where Nodes 1 and 2 are the source nodes, and the node in shaded color represents the sink. The sink wants to receive a summary of information from Nodes 1 and 2. The sensor readings generated at Nodes 1 and 2 are represented by the random variables (RV) X_1 and X_2 , respectively. Since Node 1 is a “leaf” node, Node 1 will simply transmit the raw reading X_1 to Node 2. Node 2 will combine X_1 with its own data, X_2 , by computing the summary function $g(X_1, X_2)$ which is then transmitted to the sink. We define the aggregation cost function ϕ as follows. Suppose the sensor information to be transmitted on Edge e is random variable Y . The average number of bits to be transmitted on e , or ϕ_e , is defined as (We temporarily ignore communication overheads incurred in addition to the sensor information, e.g., the packet header size. We will however take such overheads into account later when we formally define ϕ .)

$$\phi_e = H(Y), \quad (2)$$

where $H(\cdot)$ denotes the entropy function. Note that the entropy function has been also adopted as cost function in

[10, 12], and throughout this paper we will define ϕ in terms of $H(\cdot)$. The average numbers of bits transmitted on Edges 1 and 2 are, respectively, given by

$$\phi_1 = H(X_1), \quad \phi_2 = H(g(X_1, X_2)). \quad (3)$$

Suppose $g(\cdot)$ is given by *sum*. Since $H(g(X_1, X_2)) = H(X_1 + X_2) \neq H(X_1)$, the costs incurred at Edges 1 and 2, that is, ϕ_1 and ϕ_2 , are different. If we had used other types of $g(\cdot)$, such as *max*, we would have that $\phi_2 = H(\max(X_1, X_2))$ which would incur different cost from the case where $g(\cdot)$ was *sum*. In many cases we will assume symmetric sources; that is, ϕ depends only on the number of sensor readings to which $g(\cdot)$ is applied. In those cases we will treat ϕ as a function $\phi : \mathbb{Z}_+ \rightarrow \mathbb{R}_+$; that is, $\phi(m) = H(g(X_1, \dots, X_m))$ (we will also examine the cases of asymmetric sources as well). We will show that $g(\cdot)$ determines the properties of $\phi(\cdot)$ such as convexity and monotonicity, and the structure of the aggregation problem heavily depends on those properties. Hence the aggregation scheme must be designed to capture key aspects of aggregation cost functions under the given summary function. The abovementioned links among summary functions, cost functions, and optimal aggregation strategies have not been previously well studied, as we will see in Section 2 through reviewing related works.

Contributions. In this paper we investigate the minimum energy aggregation problem for several widely used summary functions. We consider two types of $g(\cdot)$. The first type is called the *summation* type which involves sums of measurements: specifically *sum*, *mean*, and *weighted sum*. The second type is called the *extreme* type which is related to the extreme statistics of the data: specifically *max* and *min*. We will use the entropy function as the measure of information rate. We show that, when $g(\cdot)$ is *sum* or *mean*, and if the source data is i.i.d., ϕ is indeed concave and increasing, *irrespective of the distribution of the source data*. This implies that one can use well-known algorithms such as the Hierarchical Matching (HM) algorithm [16] in order to approximate the optimal cost. When $g(\cdot)$ is *weighted sum* however, it is unclear how we make association between the flow aggregation problem and the cost function. Nonetheless we prove that, if the source data is independent Gaussian random variables, there exists an efficient algorithm for the problem of aggregating *weighted sum* of data with arbitrary weights.

Next we consider extreme type summary functions such as *max*. We will show that for certain distributions of source data, ϕ can be *convex* and *decreasing* in the (nonzero) number of aggregated measurements. Note that the single-sink aggregation problems for concave/increasing [16–20] or convex/increasing cost functions [21, 22] have been widely studied, however convex and decreasing ϕ has not been well studied yet. We propose a novel algorithm which effectively captures such properties of ϕ . We begin by observing that there are two aspects in cost reduction as follows. Since ϕ is convex and decreasing, ϕ decreases faster when the number of aggregated data is smaller. The intuition is that it pays to locally aggregate data among nearby sources in the *early* stages of aggregation, that is, when the number

of measurements aggregated at sensors is small. This leads us to find a low-cost *local* clustering of sources, which is a “microscopic” aspect of cost reduction. Meanwhile we need to simultaneously find a low-cost route to the sink, which must take the *global* structure of the network into account and thus is a “macroscopic” aspect of cost reduction. These are conflicting aspects and a good tradeoff point between them should be sought. To that end we propose Hierarchical Cover and Steiner Tree (HCST) algorithm. The algorithm consists of multiple stages and is designed to empirically find the best tradeoff point over the stages. We show that, by simulation, the algorithm can significantly reduce cost compared to baseline schemes such as a greedy heuristic using shortest path routing, or the HM algorithm.

Our results show that the summary function $g(\cdot)$ can significantly impact the design of aggregation schemes. However there are many choices for $g(\cdot)$: suppose for example, we would like to compute L_p norm of the vector of measurement data. Note sum and max functions which we study in this paper are in fact related such that, if the measurement data is always positive, then sum function is simply L_1 norm and max is L_∞ norm of a data vector. One could ask: what are good aggregation strategies if we take $g(\cdot)$ as a different L_p norm, say L_2 norm? We leave such questions as future work.

Paper Organization. We briefly review related work in Section 2. Section 3 introduces the model and problem formulation. Sections 4 and 5 discuss the optimal routing problem for summation and extreme type summary functions, respectively. Simulation results are presented in Section 6. Section 7 concludes the paper.

2. Related Work

In general the single-sink aggregation problem to minimize (1) is NP-hard [23], and a substantial amount of research has been devoted to designing approximated algorithms depending on certain properties of ϕ . In our case it is important to note that such properties of ϕ are determined by the choice of $g(\cdot)$. Let us briefly review the related work on the single sink aggregation problem for two types of ϕ . Most research on the single sink aggregation problem has focused on the case where ϕ is *concave* and *increasing*. Due to the concavity of ϕ , the link costs associated with the amount of aggregated data exhibit *economies of scale*, that is, the marginal cost of adding a flow at a link is cheaper when the current number of aggregated flows is greater at the link. Buy-at-bulk network design [23, 24] is based on such property of ϕ . A number of approximation algorithms have been proposed, for example, [17–19]. When ϕ is known in advance, a constant factor approximation to the optimal cost is possible [20, 25]. Even when ϕ is unknown but is concave and increasing, Goel and Estrin [16] have proposed a simple and randomized algorithm called *Hierarchical Matching* (HM). The algorithm computes minimum weight matchings of the source nodes hierarchically over stages, and outputs a tree for aggregation. HM algorithm can approximate the optimal cost by a factor logarithmic in the number of sources [16]. Nonuniform variants of this problem such that ϕ differs among the

links are also studied [26, 27] in which a polylogarithmic approximation to the optimal cost is shown to be achievable.

The case where ϕ is *convex* and *increasing* in the number of aggregated measurements has been studied in [21, 22]. Here ϕ exhibits (dis)economies of scale, that is, the marginal cost of routing a flow at a link is more expensive when a greater number of flows are aggregated at the link. Such phenomenon can be observed from many applications, such as speed scaling of microprocessors modeled by $\phi(m) = \beta m^a$ where m is the clock speed, $\beta \geq 0$ and $a \geq 1$ are constants, and $\phi(m)$ is the energy consumption at the processor. Notably the authors show that the problem can intrinsically differ from that for a concave and increasing ϕ . For example the authors show that constant-factor approximation algorithms do not exist for certain convex and increasing ϕ [21]. They nevertheless proposed a constant factor approximation algorithm for the case $\phi(m) = \beta m^a$. These results show that the single-sink aggregation problem crucially depends on certain properties of ϕ such as convexity. However, none of the above works deal with convex and decreasing ϕ which we will study in the sequel.

There have been many studies regarding the intermediate data combining in conjunction with routing in order for an efficient retrieval of the complete sensor readings. Scaling laws for achievable rates under joint source coding and routing are studied in [28]. The work [11] studies the problem of minimizing the flow costs under distributed source coding. They show that when $\phi(m)$ is linear in m , firstly applying Slepian-Wolf coding at the sources, and secondly routing coded information via shortest path tree from the sources to the sink is optimal. In [10] a single-input coding model was adopted in which the coding of information among the nodes can be done only in pairs, but joint coding of the source data from more than two nodes is not allowed. Assuming reduction in packet size is a linear function of correlation coefficient between each pair of nodes, they proposed a minimum-energy routing algorithm. The impact of spatial correlation on routing has been explored in [12]. They showed that, assuming the correlation decays over distance, it pays to form clusters of nearby nodes and aggregate data at the clusterheads. The aggregated information is then routed from clusterheads to the sink. The algorithm is shown to perform well for various correlation models. The tradeoff between integrity of aggregated information and energy consumption has been studied in [29]. Further works on in-network aggregation combined with routing include [30, 31] which propose efficient protocols for routing excessive values among sensed data. A scheme using spatially adaptive aggregation so as to mitigate traffic congestion was proposed in [32].

The above works aim at retrieving the *entire* set of data, instead of a summary, subject to certain degrees of data integrity. In our case, we design energy efficient aggregation schemes to compute the summary function $g(\cdot)$ of the sensor readings. Also in the above mentioned works, the in-network aggregation reduces cost mainly by removing correlation among the data set. In our work, by contrast, we will focus on losslessly retrieving a summary of *statistically independent* sensor readings. We assume the independence of sensor

readings because we would like to *decouple* the cost savings from removing correlation, and the savings from applying the summary function in association with aggregation strategies; we focus on the latter. Moreover, the assumption of the independence among the readings represents the “worst” case in terms of cost savings, since one cannot reduce the energy cost by removing correlations in sensor readings. In fact, the independence assumption can be valid in certain cases. For example, consider a large sensor network assuming that the sensed data is spatially correlated and the correlation decays quickly over distance. If the source nodes are sparsely deployed and thus tend to be far apart from one another, the correlation among their data can be very weak. Obviously such sparse node placement is motivated by cost efficiency: sparse placement of nodes enables us to reap as much information given a fixed number of sensor devices, assuming that the network senses a homogeneous field and the measure of information is given by the joint entropy function.

3. Model

3.1. Preliminaries. We are given an undirected graph $G = (V, E)$ where $V = \{1, 2, \dots, n\}$ and $E \subseteq V \times V$ denote the set of vertices and edges, respectively. For $u, v \in V$, $(u, v) \in E$ denotes the (undirected) edge connecting nodes u and v . For each edge in E we associate a *weight* defined by $w : E \rightarrow \mathbb{R}_+$. A weight captures the cost of transmitting unit amount of data between two nodes, for example, expenditure of transmission energy in order to compensate path loss. The set $S \subset V$ denotes the set of source nodes, that is, the nodes which generate measurement data to be reported to the sink. Also define $\eta := |S|$ where $|\cdot|$ denotes the cardinality of a set. For a source node $u \in S$, its measured data is modeled by an RV denoted by X_u . We assume that X_u 's are independent and identically distributed among the sources. The measured data is to be aggregated at the sink node denoted by $t \in V$. The nodes which are not source nodes act as relays in the aggregation process. For simplicity we will assume that any node in the network transmits data at most once during the aggregation process. Such an assumption has been made in other works such as [15]. Thus the routes for aggregation constitute a tree whose root is given by t . We refer to such tree as an *aggregation tree*. The aggregation process is performed as follows. The sources initiate transmissions. An intermediate node waits for all the data from the sources which are descendants of the node to arrive. Next the node computes the summary function of the aggregated data which is then relayed to the next hop.

In this paper a summary function is defined to be a nonnegative function denoted by $g(\cdot)$ which is a *divisible* function. Divisible functions are a class of summary functions which can be computed in a divide-and-conquer manner [1]. Divisible functions are defined as follows: given n data samples, consider a partition of the samples into sets of size k and $n - k$ denoted by $\{x_1, x_2, \dots, x_k\}$ and $\{x_{k+1}, \dots, x_n\}$, respectively. If $g(\cdot)$ is divisible, $g(g(x_1, \dots, x_k), g(x_{k+1}, \dots, x_n)) = g(x_1, x_2, \dots, x_n)$ holds for any n and k . Examples of divisible functions are sum, max and min. Particularly when $g(\cdot)$ is

divisible, the aggregation can be performed in a divide-and-conquer manner as follows. Suppose a set of data samples are aggregated at a node. If the node is a source, it applies $g(\cdot)$ to the collected samples and its own data. If the node is simply a relay, it applies $g(\cdot)$ to the aggregated data samples to obtain a summary of the samples where the summary of its aggregated data is transmitted to the next hop.

Abusing notation for the sake of simplicity, we let the function $g(\cdot)$ take a set, a vector, or their combination as its argument. For example if $g(\cdot)$ is sum, $g(x, y) = x + y$, and $g(\{x, y\}) = x + y$, and also $g(\{x, y\}, z) = x + y + z$. For some $U \subseteq V$, we define X_U as the set of RVs representing the measurements from the nodes in U ; that is, $X_U := \{u \in U \mid X_u\}$. Thus $g(X_U)$ is the aggregation function applied to the set X_U , for example, if $g(\cdot)$ is sum then

$$g(X_U) = \sum_{u \in U} X_u. \quad (4)$$

3.2. Problem Formulation. We will define the problem of minimizing communication costs as follows. There exists a sink to which the data is to be aggregated. Our goal is to find a minimum-cost aggregation tree $\mathcal{T} = (V', E') \subseteq G$ rooted at the sink. We would like to solve the following aggregation problem:

$$(P) \quad \text{Minimize} \sum_{e \in E'} w_e \phi_e, \quad (5)$$

where ϕ_e represents the average number of bits communicated over Edge e . Note that the objective of (5) has been considered in the works [10, 12] as well. We call ϕ_e as *aggregation cost function* which we define as follows.

We will use the entropy function $H(\cdot)$ as our measure of information rate similar to works [13, 14]. We assume that the average number of bits to represent random sensor measurement X is given by $H(X)$. A precise definition of the entropy function $H(X)$ depends on the nature of X : if X is a discrete RV, $H(X)$ denotes the usual Shannon entropy. If X is a continuous RV, $H(X)$ is implicitly defined to be $H(\widehat{X})$ where \widehat{X} is a discrete RV obtained by applying uniform scalar quantization to X with some quantization step size, say 2^{-b} for some integer $b > 0$. If the quantization precision is sufficiently high, it is known [33] that $H(\widehat{X}) \approx b + h(X)$ where $h(\cdot)$ denotes the differential entropy of continuous RVs. Note that a similar approximation has been made in defining the information rates for continuous RVs in [13, 14]. Hence in this paper, we will assume that continuous RV X incurs the cost of $b + h(X)$ bits where $b > 0$ is a sufficiently large parameter, and we denote such costs by $H(X) := b + h(X)$.

In addition, the measured data is transmitted as a packet in the network. Hence for each packet transmission, there is an overhead of metadata, for example, packet header. For any measurement Y , no matter how small $H(Y)$, there is always an overhead of transmitting such metadata in practice. We will assume the header length is fixed to $\alpha > 0$ bits throughout this paper. Hence the average number of bits required to send

measurement information Y per transmission over a link is given by

$$\alpha + H(Y). \quad (6)$$

For a given aggregation tree $\mathcal{T} = (V', E')$, let $p(s) \subset E'$ denote the path from a source $s \in S$ to the sink. For a given Edge $e \in E$, let $U(e) \subseteq S$ denote the set of source nodes whose aggregated measurements are transmitted over e , that is, $U(e) = \{s \in S \mid e \in p(s)\}$. The information to be communicated over Edge e is the function $g(\cdot)$ applied to the set of measurement value from $U(e)$, that is, $g(X_{U(e)})$. Hence we define the aggregation cost function as follows:

$$\phi_e = \begin{cases} 0 & U(e) = \emptyset \\ \alpha + H(g(X_{U(e)})) & \text{otherwise.} \end{cases} \quad (7)$$

We would like to solve (P) using the definition of ϕ_e given by (7). In the following sections we investigate several widely used summary functions and the associated optimal aggregation problems.

4. Aggregation Schemes for Summation-Type Summary Functions

We consider the summary functions of sum, mean, and weighted sum.

4.1. sum and mean. We first discuss the case where $g(\cdot)$ is sum. We have that

$$H(g(X_{U(e)})) = H\left(\sum_{u \in U(e)} X_u\right). \quad (8)$$

Clearly sum is a divisible function. Thus the aggregation process is as follows: a node simply applies sum function to the aggregated data, and relays the aggregated information to the next hop.

When the source data is i.i.d., we will show that there exists a randomized algorithm which finds an aggregation tree whose expected cost is within a factor of $(\log(\eta) + 1)$ of the optimal cost to (5).

Proposition 1. Suppose X_i 's are i.i.d. For any distribution of X_i , there exists an algorithm yielding the mean cost within a factor of $\{\log(\eta) + 1\}$ of the optimal cost of (P).

Goel and Estrin [16] studied a single-sink data aggregation problem as follows. A source generates a unit flow which needs to be routed to a sink where the flows are aggregated through a tree. Their objective is to minimize the following cost function:

$$\sum_{e \in E} w_e \psi(m_e), \quad (9)$$

where w_e is the weight on Edge e , m_e is the number of flows on Edge e and $\psi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a function that maps the total size of flow to its cost. They proposed an algorithm to minimize (9) when ψ is a canonical aggregation function defined as follows.

Definition 2 (see [16]). The function $\psi : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is called a canonical aggregation function (CAF) if it has the following property:

- (1) $\psi(0) = 0$.
- (2) $\psi(\cdot)$ is increasing.
- (3) $\psi(\cdot)$ is concave.

Their algorithm, called Hierarchical Matching (HM) [16], guarantees the mean cost to be within the factor of $\log(\eta) + 1$ of the optimal *irrespective of* ψ , provided that ψ is a CAF. As mentioned previously, since X_i 's are i.i.d., ϕ_e depends only on $m_e := |U(e)|$. Specifically we will define ϕ as follows:

$$\phi_e = \phi(m_e) = \begin{cases} \alpha + H\left(\sum_{u \in U(e)} X_u\right), & m_e > 0, \\ 0, & m_e = 0. \end{cases} \quad (10)$$

We will show that $\phi(\cdot)$ is a CAF by showing that $\phi(\cdot)$ satisfies the three properties of Definition 2. Note this implies that HM algorithm can be used to approximately solve (P), since (9) and the objective of (P) are identical.

Proof of Proposition 1. For the first property, it trivially holds that $\phi(0) = 0$. For the second property, for any two independent RVs X_1 and X_2 , it is known that $H(X_1 + X_2) \geq H(X_1)$ implying that $\phi(2) \geq \phi(1)$, that is, the sum of independent RVs always increases entropy [33], which implies that $\phi(m)$ is increasing in m . For the third property, consider the following. It is shown in [34] that the entropy of the sum of independent RVs is a submodular set function. That is, the following holds for independent RVs Y_1, Y_2 and Y_3 [34, Theorem I]:

$$H(Y_1 + Y_3) + H(Y_2 + Y_3) \geq H(Y_1 + Y_2 + Y_3) + H(Y_3). \quad (11)$$

Now consider $m + 2$ sensor measurements X_1, \dots, X_{m+2} , and make substitutions $Y_1 := X_{m+1}$, $Y_2 := X_{m+2}$, and $Y_3 := \sum_{i=1}^m X_i$ in (11). We have that

$$\begin{aligned} & H\left(X_{m+1} + \sum_{i=1}^m X_i\right) + H\left(X_{m+2} + \sum_{i=1}^m X_i\right) \\ & \geq H\left(\sum_{i=1}^{m+2} X_i\right) + H\left(\sum_{i=1}^m X_i\right). \end{aligned} \quad (12)$$

If we apply the definition of ϕ given by (10) to (12), the following holds due to symmetry:

$$\phi(m+1) + \phi(m+1) \geq \phi(m+2) + \phi(m). \quad (13)$$

Hence $\phi(m+2) - \phi(m+1) \leq \phi(m+1) - \phi(m)$ holds, or the slope is decreasing in m , which implies that $\phi(\cdot)$ is concave on the domain of integers. Thus $\phi(\cdot)$ satisfies all the properties of Definition 2, and is a CAF. This implies that, by using HM algorithm, one can achieve the expected cost which is within the factor of $1 + \log(\eta)$ of the optimal cost of (P). \square

Next we consider **mean** as the summary function. Note that **mean**, as well as **weighted sum** considered in the next section, are not divisible functions in general. We will nevertheless show that the problem for those summary functions can be reduced to **sum** problem as follows. Suppose every source node is aware of the total number of the sources, that is, η . In our scheme every source simply scales its measurement by η^{-1} prior to transmission, that is, Source i transmits $\eta^{-1}X_i$, then such scaled measurements are aggregated in a similar way as the **sum** problem. The average number of bits transmitted over Edge e can be written as $\alpha + H\{\sum_{u \in U(e)} ((1/\eta)X_u)\}$. Since $((1/\eta)X_u)$'s are i.i.d., for the minimum cost aggregation problem for **mean** we can use the same algorithm as that used for **sum**, for example, HM algorithm.

4.2. weighted sum. Next we consider the case where $g(\cdot)$ is **weighted sum** as follows. We assign arbitrary weights a_i , $i \in S$, to the source nodes. The goal of the sink is to compute $\sum_{i \in S} a_i X_i$. Our method of aggregation is similar to that for the case of **mean**, that is, Source i scales its measurement by a_i , then transmits $a_i X_i$ where the aggregation process is the same as that for **sum**. However the effective source data $a_i X_i$ seen by the network is no longer i.i.d., unless a_i 's are identical for all $i \in S$. The aggregation cost function is given by

$$\phi_e = \alpha + H\left(\sum_{i \in U(e)} a_i X_i\right). \quad (14)$$

The difficulty lies in it is difficult to associate a “flow” with the source data $a_i X_i$ due to asymmetry, that is, the problem is no longer a flow optimization. Moreover, it is easily seen that (14) is not a CAF in general. Thus we restrict our attention to a specific distribution of X_i . We will show that, if X_i are independent Gaussian RVs, the problem for **weighted sum** is indeed a single-sink aggregation problem with concave costs, and there exist algorithms similar to HM algorithm which have good approximation ratio. Specifically we prove that our problem is equivalent to the single-sink aggregation/flow optimization problem with *nonuniform* source demands.

Proposition 3. Suppose $X_i \sim N(\mu_i, \sigma_i^2)$, and X_1, \dots, X_η are independent. Let $g(\cdot)$ be **weighted sum** with arbitrary weights a_1, \dots, a_η . For sufficiently large b , there exists an algorithm yielding the mean cost within a factor of $\{\log(\eta) + 1\}$ of the optimal cost of (P).

Proof. Consider the information communicated over Edge e denoted by Y :

$$Y = \sum_{i \in U(e)} a_i X_i. \quad (15)$$

Since X_i 's are independent Gaussian RVs, Y is also Gaussian with variance σ_e^2 where $\sigma_e^2 := \sum_{i \in U(e)} a_i^2 \sigma_i^2$. Thus the differential entropy of Y is given by

$$h(Y) = \frac{1}{2} \log \left(2\pi e \sum_{i \in U(e)} a_i^2 \sigma_i^2 \right). \quad (16)$$

We observe that, from (16), we can treat $a_i^2 \sigma_i^2$ as the “flow” generated by Source i , and the sum of flows at Edge e incurs the entropy cost as in (16). Specifically we will make the following definitions:

$$f_i := a_i^2 \sigma_i^2, \quad i \in S, \quad (17)$$

$$f^* := \min_{i \in S} f_i, \quad (18)$$

$$\psi(x) := \alpha + b + \frac{1}{2} \log(2\pi e x), \quad x \geq 0, \quad (19)$$

$$\phi(x) := \begin{cases} \psi(x), & x \geq f^*, \\ \frac{\psi(f^*)}{f^*} x, & 0 \leq x < f^*. \end{cases} \quad (20)$$

Here f_i represents the (unsplittable) flow demand generated by Source i , and f^* denotes the minimum demand. Hence under a flow routing scheme, the total amount of flow at Link e is given by $\sum_{i \in U(e)} f_i$. Then from (16), the associated communication cost incurred at Link e is given by $\psi(\sum_{i \in U(e)} f_i)$ bits, that is, $\psi(\cdot)$ represents the information rate of a flow aggregated at Link e . Unlike the previously defined cost functions, ψ is no longer a function of the number of sources on a link, but instead the function of the amount of flow on that link. Finally we define the aggregation cost function ϕ in terms of ψ as in (20) in order to meet the concavity condition for ϕ as follows: ϕ is essentially identical to ψ , and if

$$b \geq \frac{1}{2} \log \left(\frac{1}{2\pi e f^*} \right), \quad (21)$$

one can show that $\phi(x)$ is concave and increasing for all $x \geq 0$. Hence under the condition (21), ϕ is an increasing concave function of the total flow aggregated on a link. In that case we can use the algorithm proposed by Meyerson et al. [19] which essentially extends the HM algorithm to the problems with nonuniform source flow demands, and can approximate the optimal cost by a factor of $\log(\eta) + 1$ on average. \square

In summary, the key question was whether (P) can be cast as a flow aggregation problem, if $g(\cdot)$ is **weighted sum**. In general, it is difficult to make such association due to asymmetry; however, we revealed that such formulation is possible for independent Gaussian sources.

4.3. Discussions. Note that some properties regarding X_i 's such as the submodularity relation in (11), used to show that ϕ is a CAF rely heavily on the independence of X_i 's. When X_i 's are correlated, we can find examples of ϕ which are not CAF for the summary function of **sum** as follows. Let X_1 and X_2 be jointly Gaussian with the same marginal given by $N(0, 1)$ with $E[X_1 X_2] = \rho$. Then $X_1 + X_2$ is distributed according to $N(0, 2(1 + \rho))$, thus we have that, if $\rho < -0.5$, then

$$\begin{aligned} h(X_1 + X_2) &= \frac{1}{2} \log(2\pi e \cdot 2(1 + \rho)) \\ &< \frac{1}{2} \log(2\pi e) = h(X_1). \end{aligned} \quad (22)$$

Thus the entropy function does not satisfy the second condition of Definition 2, that is, the increasing property, as a CAF. Hence for arbitrarily correlated sources, presumably few meaningful arguments can be made on optimal aggregation problems, even for simple summary functions such as *sum*.

The discussion so far enables us to deal with more general objective functions extended from (P). Consider a function $\gamma : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ which is concave and increasing. We now define communication overhead on an edge as the function γ of the average number of bits transmitted over the edge. Namely, we consider the following extension of (P):

$$(\mathbf{P}') \quad \text{Minimize} \sum_{\mathcal{T}=(V',E') \subseteq G} w_e \gamma(\phi_e). \quad (23)$$

Consider (\mathbf{P}') for the summary function *sum* for i.i.d. sources and *weighted sum* for independent Gaussian sources. Note that the composition of two concave and increasing functions is also concave and increasing [35]. Thus $\gamma(\phi_e)$ is a concave and increasing function of the amount of flows at an edge, and thus is a CAF. Hence HM algorithm can be used to approximate (\mathbf{P}') .

5. Aggregation Schemes for Extreme-Type Summary Functions

5.1. Case Study. In this section we consider summary functions regarding the extreme statistics of measurements, that is, *max* or *min*. We will first investigate the entropy of the extreme statistics of a set of RVs. Consider m measurements denoted by X_i , $i = 1, \dots, m$. Since $(\max_{1 \leq i \leq m} X_i) = -\{\min_{1 \leq i \leq m} (-X_i)\}$, we will focus only on *max* without loss of generality. It is easily seen that *max* function is divisible, thus the aggregation process is similar to that for *sum*: a node simply applies *max* function to the aggregated data. For example, suppose a node receives data given by X_1, \dots, X_m . The node simply computes $(\max_{i=1, \dots, m} X_i)$ and forwards it to the next hop.

For extreme-type summary functions, we will show that ϕ is in general *not* a CAF. In particular we consider several cases of practical importance.

Case 1 (Gaussian RVs). We consider the problem of retrieving the maximum of i.i.d. Gaussian RVs. We assume that $X_i \sim N(0, 1)$ for $i \in S$ where we again assume that $\phi(m) = \alpha + b + h(\max_{i=1, \dots, m} [X_i])$ for $m \geq 1$ and some constant b . We provide a numerical evaluation of $h(\max_{i=1, \dots, m} [X_i])$ on the left of Figure 2. We observe that $\phi(m)$ is strictly *convex* and *decreasing* in m for $m \geq 1$, thus ϕ is not a CAF.

Case 2 (Extreme data retrieval problem). We consider the problem of *extreme data retrieval* defined as follows. Assume that a source node $i \in S$ measures some physical quantity which is distributed according to a continuous RV Y_i . We assume Y_i 's are independent but not necessarily identically distributed. Suppose with some probability Y_i is equal to a large number, which indicates an “abnormal” event. An important application of sensor networks is to detect the

maximum *abnormality* among the measurements. The abnormality is defined as how far a sensor's measurement has deviated from its usual statistics as follows. Let us denote the cumulative distribution function (CDF) of Y_i by $F_i(\cdot)$ or $\mathbb{P}(Y_i \leq y) = F_i(y)$, $i \in S$. Consider realizations of Y_1, \dots, Y_η given by y_1, \dots, y_η . We will quantify the abnormality at Source i in terms of how *unlikely* the measurement y_i is: specifically the goal of the sink is to retrieve $\min_{i \in S} [\mathbb{P}(Y_i > y_i)]$, or alternatively,

$$\max_{i \in S} [F_i(y_i)], \quad (24)$$

thus the abnormality of y_i is defined by $F_i(y_i)$. Let $X_i = F_i(Y_i)$. We will assume that the nodes transmit and aggregate X_i instead of Y_i , and the goal of the sink is to retrieve $\max_{i \in S} \{X_i\}$. Note since $X_i = F_i(Y_i)$ is the RV evaluated at its distribution function, one can show that X_i 's are i.i.d. RVs uniformly distributed on $[0, 1]$. Thus the problem reduces to an optimal aggregation problem retrieving *max* of i.i.d. uniform RVs.

We will show that ϕ associated with the extreme data retrieval problem is *convex* and *decreasing* function when the number of aggregated measurements is greater than or equal to 2. Suppose the data aggregated at a node is given by X_1, \dots, X_m and define $Z_m := \max_{i=1, \dots, m} [X_i]$. As previously we assume that the node requires on average $\phi(m) = \alpha + b + h(Z_m)$ bits to transmit Z_m .

Proposition 4. Consider the extreme data retrieval problem. The aggregation cost function $\phi(m)$ is convex and decreasing for $m \geq 2$.

Proof. Since Z_m is the maximum of m i.i.d. uniform RV's, the CDF of Z_m denoted by $F_{Z_m}(\cdot)$ is given by

$$F_{Z_m}(z) = z^m. \quad (25)$$

Thus the probability density function (pdf) of Z_m denoted by f_{Z_m} is given by mz^{m-1} . If we compute $h(Z_m)$,

$$\begin{aligned} h(Z_m) &= - \int_0^1 f_{Z_m}(z) \log \{f_{Z_m}(z)\} dz \\ &= - \int_0^1 mz^{m-1} \log(mz^{m-1}) dz = -\log(m) + \frac{m-1}{m}. \end{aligned} \quad (26)$$

Thus

$$\phi(m) = \begin{cases} \alpha + b - \log(m) + 1 - m^{-1}, & m \geq 1, \\ 0, & m = 0. \end{cases} \quad (27)$$

By regarding m as a continuous variable, we have that, for $m \geq 1$,

$$\frac{d\phi}{dm} = -\frac{1}{m} + \frac{1}{m^2}, \quad \frac{d^2\phi}{dm^2} = \frac{1}{m^2} - \frac{2}{m^3} = \frac{1}{m^2} \left(1 - \frac{2}{m}\right). \quad (28)$$

Clearly $\phi(m)$ is decreasing for $m \geq 1$, and since its second order derivative is nonnegative for $m \geq 2$, $\phi(m)$ is convex for $m \geq 2$. \square

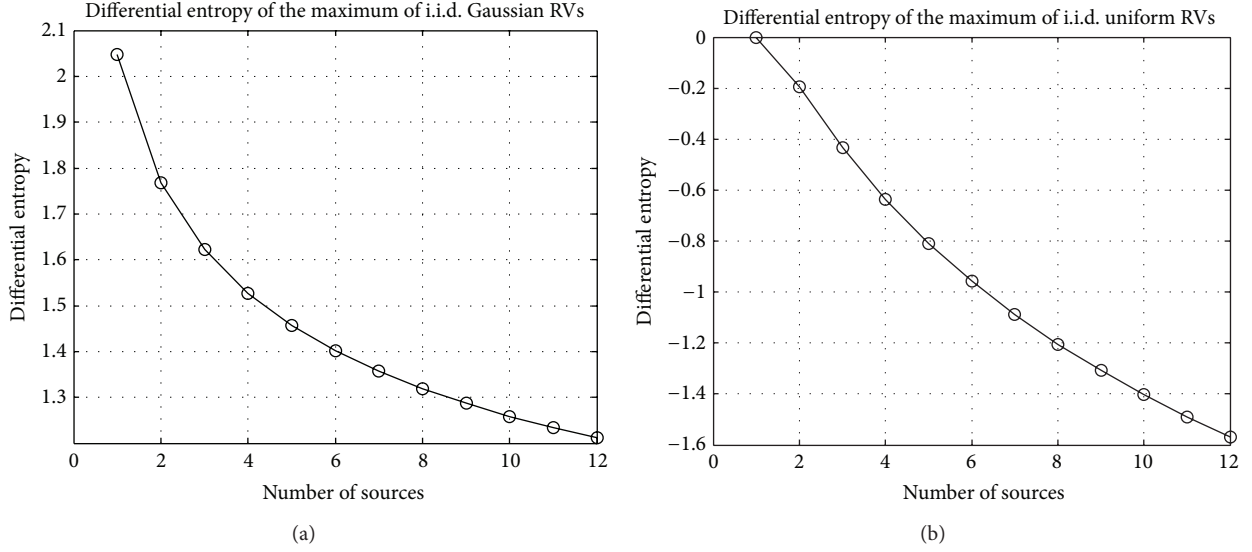


FIGURE 2: The differential entropy of the maximum of a set of i.i.d. RVs distributed according to an RV X . On the left, $X \sim N(0, 1)$, and on the right, X is uniform on $[0, 1]$.

On the right of Figure 2 the plot of $h(Z_m)$ is shown. Note $h(Z_m)$ is strictly convex for $m \geq 2$, but overall appears to be approximately convex. Note that $h(Z_m)$ is nonpositive, thus one could select a sufficiently large b such that $\phi(\eta) = \alpha + b + h(Z_\eta) \geq 0$, so that $\phi(m) \geq 0$ for all $1 \leq m \leq \eta$.

In general, for a convex and decreasing ϕ , (P) is clearly NP-hard since the problem contains the Steiner tree problem as a special case. In the following section we present a novel algorithm which captures key properties of convex and decreasing ϕ . Later we show by simulation the algorithm effectively achieves low cost.

5.2. Algorithm for Convex and Decreasing Aggregation Cost Functions

5.2.1. Motivation. Before we describe our algorithm we present the motivation behind the algorithm. An important observation for the data aggregation problems was made in [25] for *concave* and *increasing* ϕ . They proposed a “hub-and-spoke” model for so-called *facility location problem*. The idea is that when ϕ is concave and increasing, one should first aggregate flows to some “hubs,” then route the aggregated flow from the hubs to the sink at the minimum cost; this is done by building an approximately optimal Steiner tree where the hubs (facility locations) are the Steiner nodes. The rationale is that, once multiple flows are aggregated at hubs, the cost of routing them collectively to the sink is cheaper than routing the sources’ flows separately, due to the concavity of ϕ . We observe two aspects in such hub-and-spoke schemes. Firstly by local aggregation of flows at hubs we aim at greedily reducing costs based on local information, which we view as the *microscopic* approach to reduce cost. Secondly by building an approximately optimal Steiner tree with respect to the hubs and the sink, we take the *global* network structure into account, which can thus be seen as

the *macroscopic* aspect for cost reduction. Hence there exists a tradeoff between microscopic and macroscopic aspects of the cost reduction. A similar observation on such tradeoff was made in [12]. However our key question is that, how do we achieve an optimal tradeoff between those aspects for a *convex* and *decreasing* ϕ ?

Consider the three examples of aggregation cost functions denoted by ϕ_1 , ϕ_2 , and ϕ_3 which are decreasing and convex for $m \geq 1$ as shown in Figure 3. In case of ϕ_1 , we see that ϕ_1 is flat for $m \geq 1$, that is, the average number of bits communicated over a link is constant irrespective of the number of flows passed through it. Thus, the minimum cost routing problem reduces to a Steiner tree problem, in which case a completely “macroscopic” solution is optimal. In case of ϕ_2 , we see that ϕ_2 decreases slowly in m . Thus, the more number of flows merges at a link, it takes the less number of bits to transmit the merged information. Suppose we use the hub-and-spoke scheme to aggregate flows in a local manner. The amount of aggregated flows at a hub is at least 2: note that however, ϕ_2 is approximately “flat” for $m \geq 2$. This implies that, once more than two flows are aggregated, the benefits from further local flow aggregation will be negligible. Hence the optimal routing problem from the hubs to the sink approximately reduces to the Steiner tree problem! Thus one could expect that local aggregation (microscopic approach) followed by an optimal Steiner tree construction (macroscopic approach) would yield a good solution. Now let us consider ϕ_3 . The overall rate of decrease of ϕ_3 is higher than that of ϕ_2 . It appears that when the number of aggregated flows is significantly high, for example, m is greater than 6, ϕ_3 becomes effectively “flat.” This suggests that, one should keep aggregating flows until sufficient amount of flows, say 6, is aggregated, that is, the microscopic cost reduction should be applied for *multiple times in a hierarchical manner*, then build an optimal Steiner tree with respect to the aggregated sources, that is, applying macroscopic reduction.

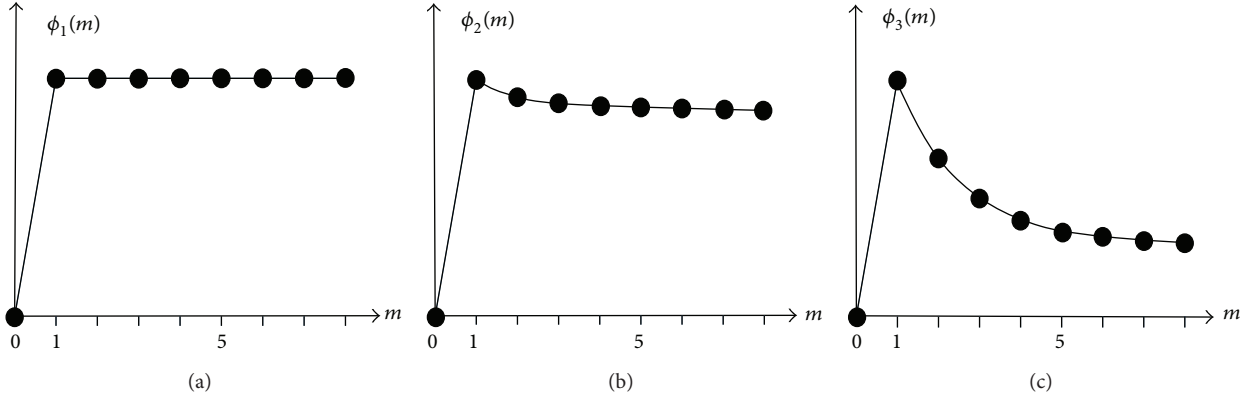


FIGURE 3: Aggregation cost functions which are convex and decreasing for $m \geq 1$.

The example provides us with some insights. Since $\phi(m)$ is convex decreasing, the marginal benefit of local aggregation is large for small m but decreases with increasing m . In other words, when m is small, that is, in the early stages of the overall aggregation process, one should focus on low-cost local aggregation in order to benefit from high rate of decrease of $\phi(m)$ for small m . Meanwhile, once a large number of flows are aggregated, it pays to perform macroscopic cost reduction from there on by building the optimal Steiner trees since ϕ becomes more “flat” with increasing m . This suggests that there exists a tradeoff point at which such microscopic and macroscopic reduction are optimally balanced. Unfortunately it is difficult to know such a tradeoff point in advance. The proposed algorithm not only exploits both the microscopic and macroscopic aspects of cost reduction for a convex and decreasing ϕ , but also empirically searches for the optimal tradeoff point. Details are presented in the following section.

5.2.2. Outline. An outline of the proposed algorithm is presented as follows. The algorithm consists of multiple stages. A hub-and-spoke problem (or facility location problem) is approximately solved at each stage. The flows from source nodes are merged at the hubs. The hubs at the present stage become the source nodes in the next stage, that is, the flows are merged hierarchically. Instead of solving complex facility location problem, we find a *minimum weight edge cover* (MWECC) on the source nodes at each stage as a simple approximation. The rationale is that we would like to cluster sources for local aggregation at low costs, and by definition the MWECC incurs low cost in doing that. MWECC consists of multiple connected components, each of which is a tree. For each connected component we select a source as a hub and call it a *center node* (details on the selection of center nodes are provided later). The flows in that component is aggregated at the center node.

At each stage, once the center nodes are determined, we build an approximately optimal Steiner tree with respect to the center nodes and the sink. We use algorithm in [36] for the Steiner tree construction. Their algorithm provides the best

known ρ_S -approximation for Steiner tree problem where $\rho_S \approx 1.39$.

Each stage outputs an aggregation tree. The output tree at Stage i is the *union* of the paths from all the hierarchical aggregations found up to Stage i and the Steiner tree built at Stage i . Namely, the output tree at Stage i is a combination of i consecutive hierarchical aggregations (microscopic cost reduction) and a Steiner tree with respect to the sink and Stage i hubs (macroscopic cost reduction).

Hence, over the stages, the algorithm progressively changes the balance between microscopic and macroscopic aspects of cost reduction in the output trees. Roughly speaking, the output trees from later stages are more biased towards the microscopic aspect. After the stages are over, we pick the tree with the minimum cost among the output trees. As a result the algorithm empirically searches for the point of the “best” balance between the two aspects of cost reduction over the stages. Hence one could expect that our algorithm will work well for any convex and decreasing ϕ .

5.2.3. Algorithm Description. We present a formal description of the proposed algorithm followed by an explanation of further details. For given aggregation tree $\mathcal{T} \subseteq G$, let $c(\mathcal{T})$ denote the total energy cost associated with \mathcal{T} , as in the objective of (P).

Hierarchical Cover and Steiner Tree (HCST) Algorithm

Begin Algorithm

- (1) (Metric completion of G) If G is not a complete graph, perform a metric completion of G to yield a complete graph. Namely, if there exist any pair of vertices without an edge, create an edge between the pair and assign the edge a weight which is the distance between the pair. The distance is measured in terms of the sum of the weights on the shortest path between the pair.
- (2) (Initialization) $i \leftarrow 0$, $S_0 \leftarrow S$.
- (3) (Initialize flows at sources) $f_u \leftarrow 1$, for all $u \in S$.

- (4) (Initial output is a Steiner tree) Jump to Step 7.
- (5) (Minimum weight edge cover) Let us denote the subgraph of G induced by S_{i-1} by G_s . Find a minimum-weight edge cover M_i in G_s . Let $C_i = (S_{i-1}, M_i)$ be the subgraph of G induced by the cover.
- (6) (Node selection) Suppose C_i has ν connected components, and denote the j th connected component of C_i by $K_j = (V_j, E_j)$ for $1 \leq j \leq \nu$. For each K_j , select a node with the maximum degree (ties are arbitrarily broken), say u_j , which is called a *center node*. K_j is a tree, and u_j becomes the root of K_j . All the flows in K_j are aggregated at u_j such that every node transmits data to its parent node after the data from its child nodes has been aggregated at the node. The total flow at u_j is updated as follows:

$$f_{u_j} \leftarrow \sum_{u \in V_j} f_u. \quad (29)$$

Remove all the noncenter nodes from S_{i-1} , and let S_i be the resulting set of source nodes.

- (7) (Steiner tree construction) Build ρ_s -optimal Steiner tree T_i^s with respect to the source nodes in S_i and the sink, using the algorithm in [36].
- (8) (Merging trees) If $i > 0$, merge all the MWECs found up to the present stage and the Steiner tree found in Step 7; that is, let

$$T_i \leftarrow T_i^s \cup \left(\bigcup_{j=1}^i C_j \right). \quad (30)$$

If $i = 0$, $T_0 \leftarrow T_0^s$. We call T_i the output tree of Stage i .

- (9) (Loop) If $|S_i| > 1$, $i \leftarrow i + 1$ and go back to Step 5. If $|S_i| = 1$, continue to Step 10.
- (10) (Tree selection) The final output is the tree T_{j^*} such that

$$j^* = \underset{j=0, \dots, i}{\operatorname{argmin}} [c(T_j)], \quad (31)$$

that is, the minimum cost tree among the output trees from all the stages.

End Algorithm

5.2.4. Comments. We explain the details of several steps in the algorithm. In Step 3 the flow variables denoted by f_u , $u \in S$, associated with the source nodes are initialized where we will track the amount of flows throughout the algorithm. In Step 6 it is natural to select a node with the maximum degree as the center node, since such node is literally a “hub.” When solving the hub-and-spoke problem at each stage, we choose to solve the MWEC problem whereas in [25] the load-balanced facility location problem is solved. An advantage of solving MWEC problem is that it is considerably simpler than load-balanced facility location problems since an MWEC problem

can be reduced to a minimum weight perfect matching problem [37]. Note that the algorithm in [25] solves the hub-and-spoke problem only once, that is, its output is analogous to the output tree from Stage 1 of our algorithm. Meanwhile HM algorithm solves minimum weight perfect matching at each stage in order to locally aggregate flows with low costs. HM algorithm solves the matching problem hierarchically until all the flows are aggregated to a single source, and the final output is the union of those matchings. Thus its final output is analogous to that from the final stage of our algorithm. In other words, the outputs of the abovementioned algorithms correspond to those from intermediate stages in our algorithm. The HIERARCHY algorithm proposed in [20] hierarchically constructs Steiner trees and solves load-balanced facility location problems, however in a way which heavily relies on the concave and increasing property of ϕ . Thus the algorithm may not be suitable for convex and decreasing ϕ .

5.3. Performance Analysis. In this section we analyze the performance of HCST algorithm. For set \mathcal{E} of weighted edges, let $\|\mathcal{E}\|$ denote the sum of its edge weights, that is, $\sum_{e \in \mathcal{E}} w_e$. For given source set Σ , let $T_s(\Sigma)$ denote the edge set of the optimal Steiner tree associated with Σ .

Proposition 5. For given network graph $G = (V, E)$, the cost achieved by HCST algorithm is higher than the optimal algorithm by a factor of at most B defined as

$$B := \frac{1}{\phi(Z)} \cdot \min_{i=0, \dots, \mathcal{J}} \left[\phi(2^i) \rho_s + \sum_{k=1}^i \phi(2^{k-1}) \rho_k \right], \quad (32)$$

where \mathcal{J} ($\mathcal{J} \leq \log_2 \eta$) denotes the stage at which HCST algorithm terminates. $\rho_s \approx 1.39$ denotes the approximation ratio for Steiner tree problem, and $\rho_i \in [0, 1]$ is the ratio of the sums of edge weights between MWEC M_i at Stage i of HCST algorithm and the Steiner tree associated with source set S , that is,

$$\rho_i = \frac{\|M_i\|}{\|T_s(S)\|}. \quad (33)$$

Also Z is defined as

$$Z := \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \eta \sum_{i=0}^{n-\lfloor \eta/2 \rfloor-1} w_{[\lfloor E \rfloor - i]}}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=0}^{n-\lfloor \eta/2 \rfloor-1} w_{[\lfloor E \rfloor - i]}}, \quad (34)$$

where $n := |V|$, and $w_{[i]}$ denotes the i th smallest value of the edge weights of G . Note that the second summation term of (32) is defined to be 0 if $i = 0$.

Proof. Denote the optimal cost by OPT. We first find a lower bound for OPT. Let T^* denote the set of edges of the optimal aggregation tree. Let us sort the amount of edge flows of T^* in increasing order, and denote them by d_i , that is, $0 < d_1 \leq d_2 \leq \dots \leq d_l$ where T^* has l edges. There are at least η nonzero flows since there are η sources, hence $d_\eta > 0$ and $l \geq \eta$ hold. In addition l is at most n , since T^* is a tree. Also it is clear that

$d_i \leq i$, for $i = 1, \dots, \eta$, and d_i is at most η for $\eta < i \leq l$. This implies that, since ϕ is decreasing, $\phi(d_i) \geq \phi(i)$, $i = 1, \dots, \eta$. Let us denote the weight of the edge that carries flow d_i by v_i . For real numbers a and b , let $a \wedge b := \min(a, b)$. We have that

$$\text{OPT} = \sum_{i=1}^l v_i \phi(d_i) \geq \sum_{i=1}^l v_i \phi(i \wedge \eta) \quad (35)$$

$$\geq \left(\sum_{i=1}^l v_i \right) \phi \left(\frac{\sum_{i=1}^l (i \wedge \eta) v_i}{\sum_{i=1}^l v_i} \right) \quad (36)$$

$$\geq \|T_S(S)\| \phi \left(\frac{\sum_{i=1}^l (i \wedge \eta) v_i}{\sum_{i=1}^l v_i} \right), \quad (37)$$

where (36) is by Jensen's inequality due to the convexity of ϕ , and (37) is from the definition of Steiner trees. Considering that ϕ is decreasing, we would like to make the argument of ϕ in (37) as large as possible in order to find a lower bound for OPT. Hence we would like to maximize $\lambda(v_1, \dots, v_l)$ defined as

$$\lambda(v_1, \dots, v_l) := \frac{\sum_{i=1}^l (i \wedge \eta) v_i}{\sum_{i=1}^l v_i}, \quad (38)$$

where v_i , $i = 1, \dots, l$ are chosen from the edge weights of G . For the purpose of maximizing (38), we will assume $v_1 \leq v_2 \leq \dots \leq v_l$ WLOG, because over all possible permutations $\pi(1), \pi(2), \dots, \pi(l)$ of $\{1, 2, \dots, l\}$, $\sum_i (i \wedge \eta) v_{\pi(i)}$ is maximized when $v_{\pi(1)} \leq v_{\pi(2)} \leq \dots \leq v_{\pi(l)}$.

We first observe that $\lambda(\cdot)$ is decreasing in $v_1, \dots, v_{\lfloor \eta/2 \rfloor}$, since if $k \leq \lfloor \eta/2 \rfloor$, we have that

$$\begin{aligned} \frac{\partial \lambda}{\partial v_k} &= \frac{k \sum_i v_i - \sum_i (i \wedge \eta) v_i}{(\sum_i v_i)^2} \\ &= \frac{\sum_{j=1}^{k-1} (k-j)(v_j - v_{2k-j}) + \sum_{j=2k}^l (k - (j \wedge \eta)) v_j}{(\sum_i v_i)^2} \\ &\leq 0. \end{aligned} \quad (39)$$

Hence $\lambda(\cdot)$ can be maximized over $v_1, \dots, v_{\lfloor \eta/2 \rfloor}$ by choosing $\lfloor \eta/2 \rfloor$ smallest weights from the edge weights of G , that is, by letting $v_i = w_{[i]}$ for $i = 1, \dots, \lfloor \eta/2 \rfloor$.

Next we would like to derive an upper bound for $\lambda(w_{[1]}, \dots, w_{\lfloor \eta/2 \rfloor}, v_{\lfloor \eta/2 \rfloor+1}, \dots, v_l)$ as follows:

$$\begin{aligned} \lambda(w_{[1]}, \dots, w_{\lfloor \eta/2 \rfloor}, v_{\lfloor \eta/2 \rfloor+1}, \dots, v_l) \\ = \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \sum_{i=\lfloor \eta/2 \rfloor+1}^l (i \wedge \eta) v_i}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=\lfloor \eta/2 \rfloor+1}^l v_i} \end{aligned} \quad (40)$$

$$\leq \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \eta \sum_{i=\lfloor \eta/2 \rfloor+1}^l v_i}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=\lfloor \eta/2 \rfloor+1}^l v_i} \quad (41)$$

$$\leq \frac{\sum_{i=1}^{\lfloor \eta/2 \rfloor} i w_{[i]} + \eta \sum_{i=0}^{n-\lfloor \eta/2 \rfloor-1} w_{[|E|-i]}}{\sum_{i=1}^{\lfloor \eta/2 \rfloor} w_{[i]} + \sum_{i=0}^{n-\lfloor \eta/2 \rfloor-1} w_{[|E|-i]}}. \quad (42)$$

For inequality (42), we used the fact that (41) is increasing in $\sum_{i=\lfloor \eta/2 \rfloor+1}^l v_i$, hence we chose $l = n$ and the largest possible weights $w_{[|E|]}, w_{[|E|-1]}, \dots$ for $v_{\lfloor \eta/2 \rfloor+1}, v_{\lfloor \eta/2 \rfloor+2}, \dots$, in order to maximize $\sum_{i=\lfloor \eta/2 \rfloor+1}^l v_i$. From (42), we obtain $\lambda(v_1, \dots, v_l) \leq Z$. Hence from (37), we obtain

$$\text{OPT} \geq \|T_S(S)\| \phi(Z). \quad (43)$$

Now let us consider the cost of output tree at Stage i of HCST algorithm, or $c(T_i)$. Recall that in HCST algorithm, S_i denotes the source set at Stage i , and T_i denotes the output tree at Stage i . The cost of T_i is divided into (i) the cost incurred by hierarchical MWECs M_1, \dots, M_i , and (ii) the cost of ρ_S -approximate Steiner tree T_i^s associated with S_i . Hence

$$c(T_i) = \sum_{k=1}^i \sum_{e \in M_k} w_e \phi(d(e)) + \sum_{e \in T_i^s} w_e \phi(d(e)), \quad (44)$$

where $d(e)$ denote the amount of flow at Edge e under HCST algorithm. Note that, the amount of flow in the network at Stage i is at least 2^{i-1} , since the flows are agglomerated through MWECs at every stage. Since $\phi(\cdot)$ is decreasing, the first summation of (44) is at most

$$\sum_{k=1}^i \phi(2^{k-1}) \sum_{e \in M_k} w_e = \sum_{k=1}^i \phi(2^{k-1}) \|M_k\|. \quad (45)$$

Note that the first summation of (44) is 0 for Stage 0. As for the second summation of (44),

$$\sum_{e \in T_i^s} w_e \phi(d(e)) \leq \phi(2^i) \|T_i^s\| \quad (46)$$

$$\leq \rho_S \phi(2^i) \|T_S(S_i)\| \quad (47)$$

$$\leq \rho_S \phi(2^i) \|T_S(S)\|. \quad (48)$$

Inequality (48) is due to $S_i \subseteq S$; specifically, the Steiner tree for S is a tree that spans S_i , hence by definition, the sum of edge weights of $T_S(S_i)$ is no more than that of the Steiner tree associated with S .

In conclusion, we have that, from (43), (45) and (48),

$$\begin{aligned} c(T_i) &\leq \rho_S \phi(2^i) \|T_S(S)\| + \sum_{k=1}^{i-1} \phi(2^{k-1}) \|M_k\| \\ &\leq \frac{\text{OPT}}{\phi(Z)} \left[\rho_S \phi(2^i) + \sum_{k=1}^i \phi(2^{k-1}) \frac{\|M_k\|}{\|T_S(S)\|} \right]. \end{aligned} \quad (49)$$

Since the cost of HCST algorithm is $\min_{i=0, \dots, \mathcal{T}} c(T_i)$, the proposition is proved. \square

An interpretation for ratio B in (32) is as follows: the first term in the bracket of B represents a bound on the macroscopic cost associated with the Steiner tree approximation. The second term in the bracket of B is a bound on the cost associated with the hierarchical aggregation of flows, that is, the microscopic cost reduction. Clearly we have

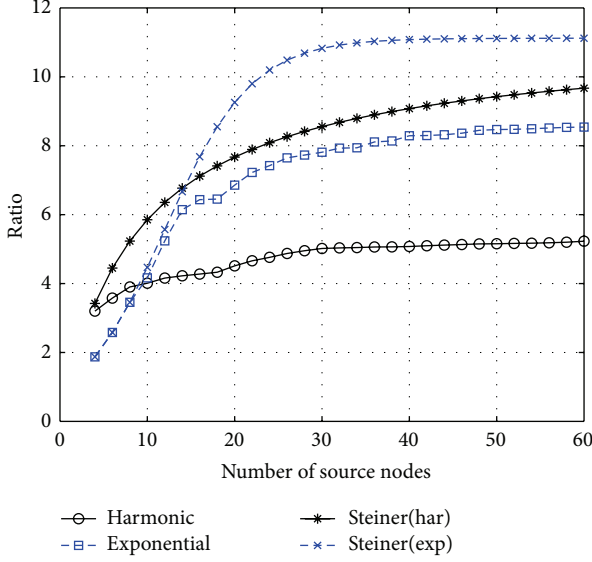


FIGURE 4: Performance bounds under varying number of sources.

that $\|M_1\| \geq \|M_2\| \geq \dots$, due to $S_1 \supseteq S_2 \supseteq \dots$, thus ρ_1, ρ_2, \dots , is a decreasing sequence where $0 \leq \rho_i \leq 1$, for all i . The progressive cost reduction due to hierarchical flow aggregation is reflected in ρ_1, ρ_2, \dots . As in (32), B is the minimum of $\mathcal{F} + 1$ numbers, each of which contains a weighted sum of $\phi(\cdot)$ in different combination of weights ρ_i . Hence B represents the empirical minimum of different degrees of tradeoff between microscopic and macroscopic cost reduction.

Next we discuss constant Z in (34). Firstly observe that $Z \leq \eta$; the first summation of the numerator of (34) is at most $\eta \sum_{i=1}^{\lfloor \eta/2 \rfloor} v_i$, in which case the first term of (34) is at most η . Note that a naive upper bound for $\lambda(v_1, \dots, v_n)$ is simply η , yielding a lower bound $\text{OPT} \geq \|T_S(S)\| \phi(\eta)$; however we observe that our bound (43) improves such a bound since $\phi(Z) \geq \phi(\eta)$.

B can be numerically computed for a given graph, and in the next section we provide numerical examples of B . We also apply HCST algorithm to a specific graph as an example.

5.4. Illustrating Examples. In this section we consider a simple convex and decreasing ϕ . As previously the packet header length is α bits, and we assume that the maximum packet size is 10 times the header length, that is, 10α . We will accordingly consider $\phi(m)$ which is convex and decreasing for $m \geq 1$ of the following form:

$$\phi(m) = \begin{cases} 9\alpha m^{-1} + \alpha, & m \geq 1, \\ 0, & m = 0. \end{cases} \quad (50)$$

Clearly $\alpha < \phi(m) \leq 10\alpha$ holds for $m \geq 1$.

Figures 4 and 5 show the numerical examples of the performance bound B . B is computed and averaged over randomly generated graphs of uniformly distributed nodes in a square area. In Figure 4, network size n is fixed to 200, and B is plotted against the number of source nodes η . We consider

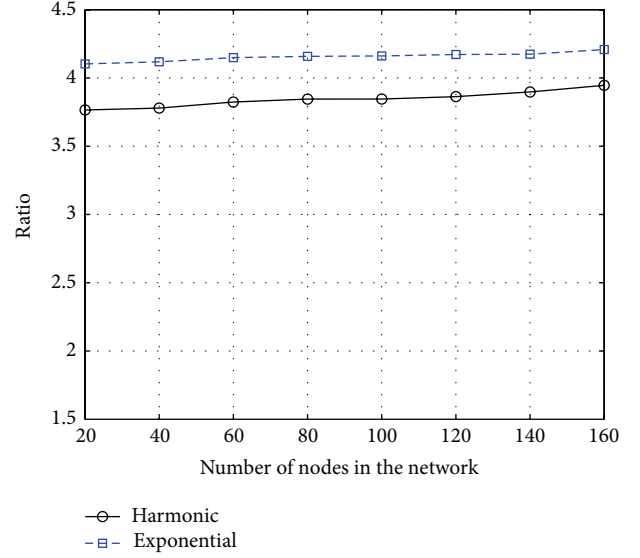


FIGURE 5: Performance bounds under varying network sizes.

two types of cost functions: the curve labelled “harmonic” represents the cost function (50) in which $\phi(\cdot)$ decreases as a harmonic sequence. The curve labelled “exp” corresponds to the case where the term m^{-1} in (50) is replaced by $\exp(-\delta(m-1))$ where the parameter $\delta > 0$ controls the decay rates of the cost function. We set $\delta = 0.2$ in this example. In addition, we compare B with a simple analytical bound; suppose we build a ρ_S -approximate Steiner tree based on S . The cost under that tree is at most $\rho_S \|T_S(S)\| \phi(1)$. By combining that cost with (43), we obtain a simple approximation ratio of $\rho_S \phi(1) / \phi(Z)$ for the approximately optimal Steiner tree. In Figure 4, the plots of such bounds based on ρ_S -approximate Steiner tree are added for both harmonic and exponential cost functions, and are labelled as “Steiner(har)” and “Steiner(exp),” respectively. We observe that B provides improved bounds as compared to those based on ρ -approximate Steiner tree. In Figure 5, B is plotted against varying n under the aforementioned harmonic and exponential cost function where we fixed δ to 10. In Figures 4 and 5, we observe that B eventually becomes nearly constant, or increases very slowly at most, even if the system size grows. Hence we conclude that B provides an approximation ratio which remains effectively constant irrespective of the system size.

Next we present an example of the application of the HCST algorithm to a specific graph. An example of G is given in Figure 6(a). G consists of $n = 10$ nodes where Node 1 is the sink, that is, $t = 1$. There are four source nodes: $S = \{2, 3, 4, 5\}$ where the sources are depicted in a shaded color. Each source generates 1 unit of data. We will again consider convex and decreasing $\phi(\cdot)$ given by (50), and assume $\alpha = 1$. Figure 6(b) shows the output of Stage 0 or T_0 which is an approximately optimal Steiner tree. Figure 7 shows the MWECs over the stages. Figure 7(a) shows the metric completion of the subgraph induced by S . Figure 7(b) shows the MWEC at Stage 1. Node 4 and 5 became the center nodes as emphasized in the figure. Figure 7(c) shows the MWEC and the center node at Stage 2.

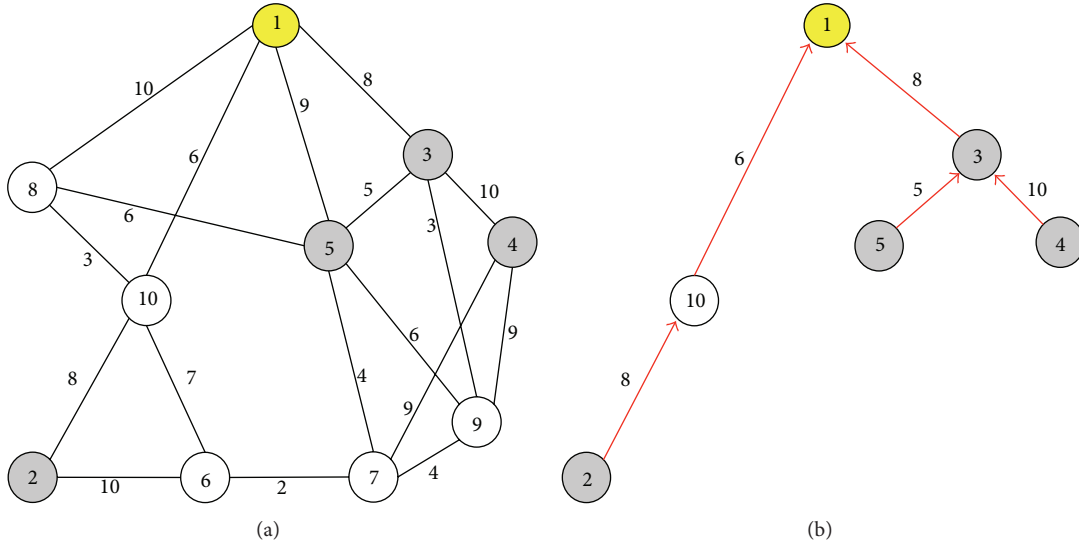
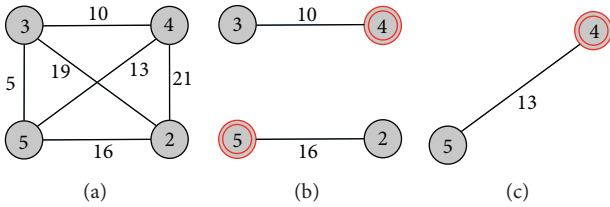
FIGURE 6: (a) G in the example. (b) The output T_0 from Stage 0.

FIGURE 7: (a) A complete graph of the sources. (b) MWEC from Stage 1. (c) MWEC from Stage 2.

Figure 8(a) shows the full paths of the MWEC at Stage 1, that is, that in Figure 7(b), in G . By building an approximately optimal Steiner tree T_1^s associated with $\{1, 4, 5\}$ and taking the union of T_1^s and C_1 as in Step 8, we get T_1 as in Figure 8(b). Similarly Figure 9 demonstrates Stage 2 of the algorithm. The full paths for the MWEC from Figure 7(c) in G are shown in Figure 9(a). Note that Node 4 is selected as the center node, and the output from Stage 2 or T_2 is shown in Figure 9(b). Let us compare the energy costs from all the stages. For T_0 , a total of three flows pass through the link between Node 1 and 3, while the flow on the other links is simply 1. Thus, the cost of T_0 from Stage 0 is given by

$$\begin{aligned} c(T_0) &= 8 \times \phi(3) + (8 + 6 + 5 + 10) \times \phi(1) \\ &= 8 \times 10 + 29 \times 4 = 322. \end{aligned} \quad (51)$$

Similarly, we have that

$$c(T_1) = 314.8, \quad c(T_2) = 275.5. \quad (52)$$

Thus the final output of HCST is T_2 with the final cost of 275.5. Note that in this example, the Shortest Path Tree (SPT) heuristic incurs the energy cost of 374.

Next consider $\phi(m)$ such that

$$\phi(m) = \begin{cases} 1, & m \geq 1, \\ 0, & m = 0. \end{cases} \quad (53)$$

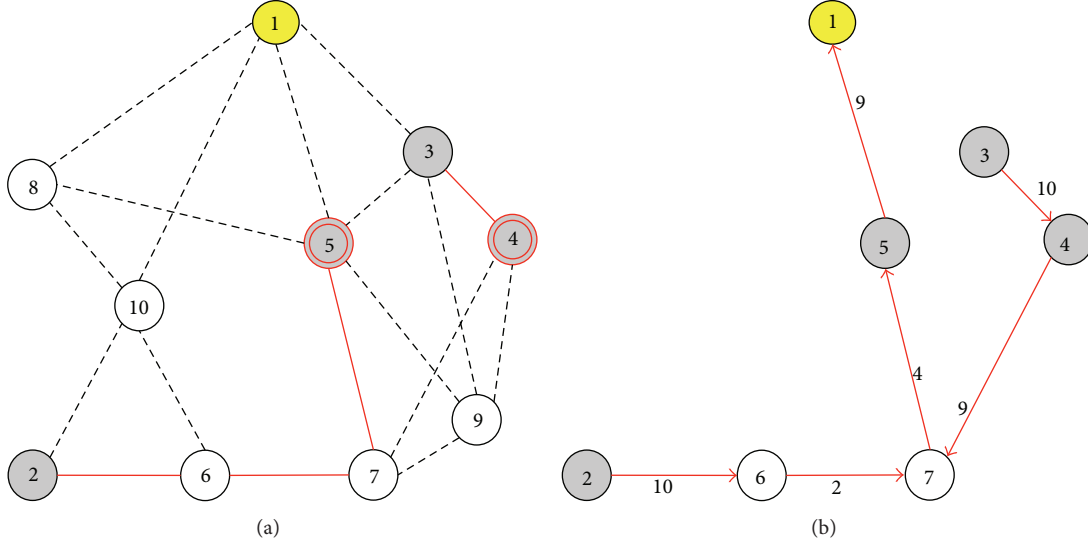
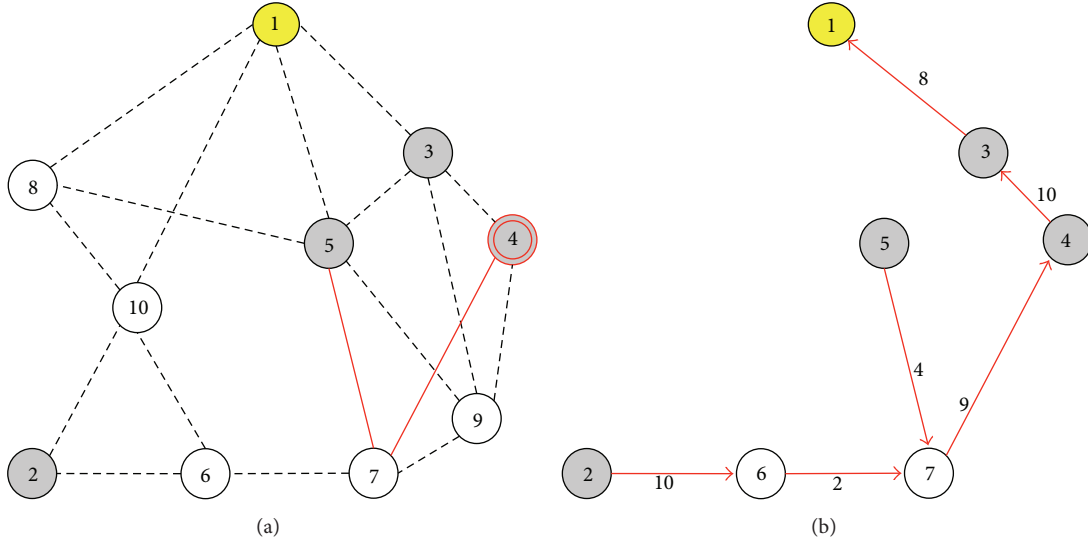
Assume that the algorithm has yielded the same T_0 , T_1 and T_2 as the previous case. Since ϕ is constant for $m \geq 1$, the problem reduces to the Steiner tree problem, thus one would expect that T_0 would perform the best since T_0 is intended to be an approximately optimal Steiner tree. The energy costs are given by

$$c(T_0) = 37, \quad c(T_1) = 44, \quad c(T_2) = 43; \quad (54)$$

thus indeed the HCST algorithm will output T_0 as the best solution with cost 37, whereas the SPT heuristic will yield the energy cost of 41. This demonstrates that our algorithm can effectively deal with various types of convex and decreasing aggregation cost functions. In the following section we will evaluate the performance of the HCST algorithm by simulation.

6. Simulation

In our simulation we randomly generate G as follows. The node locations are generated independently and uniformly on a unit square. We define G as the Delaunay graph induced by the node locations. An example of G is depicted in Figure 10 for $n = 20$. As previously it is assumed that the average number of bits required to transmit the aggregated information $g(X_1, \dots, X_m)$ is approximately $a + b + h(g(X_1, \dots, X_m))$ where we set header length a to 1 and the number of quantization bits b to 3. The edge weights are randomly selected from $\{1, \dots, 10\}$ which represents the energy consumption per transmitted bit. In our simulation two types of sources are considered. The first type, called uniform type, is associated with the extreme data retrieval

FIGURE 8: (a) MWEC in G from Stage 1. (b) The output T_1 from Stage 1.FIGURE 9: (a) MWEC in G from Stage 2. (b) The output T_2 from Stage 2.

problem, that is, X_i are i.i.d. uniformly on $[0, 1]$. The second type, called Gaussian type, is associated with retrieving the maximum of Gaussian source data where $X_i \sim N(0, 1)$. The summary function $g(\cdot)$ is given by max function.

We will compare the performance of the HCST algorithm with HM algorithm [16] and SPT heuristic. Figure 11 shows the average energy consumption of the algorithms when we fix the number of sources to 8 with varying n . The energy cost shown on the left (resp. right) of Figure 11 is associated with the sources of uniform (resp. Gaussian) type. We observe that the HCST algorithm achieves lower energy costs than the SPT heuristic in both types of the sources. The gain in the energy savings by the HCST algorithm ranges 35–38% for uniform type sources and 24–25% for Gaussian type sources. Compared to HM algorithm, our algorithm reduces the energy consumption by 20–21% and 14–15% for uniform and

Gaussian type sources, respectively. HM algorithm focuses on microscopic cost reduction, which may be effective for concave and increasing cost functions, however not for convex and decreasing cost functions. Comparing SPT heuristic and HCST algorithm, we observe that the difference in the mean energy consumption of the algorithms slightly increases with n . This can be interpreted as follows: for larger networks, there is further room for improvement by HCST, for example, there are more choices for Steiner nodes and more ways to merge sources at low costs by MWEC. Thus the performance gain from the HCST algorithm relative to the SPT heuristic is expected to grow with n as shown in the simulation.

Figure 12 shows the mean energy costs with varying n where we scale the number of the sources proportional to n . Specifically in the simulation we let $\eta = n/5$, that is, one out of five nodes is a source node. In the figure we see that the HCST

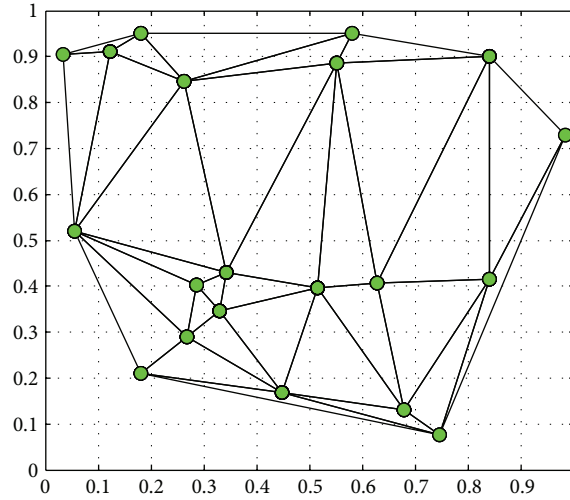
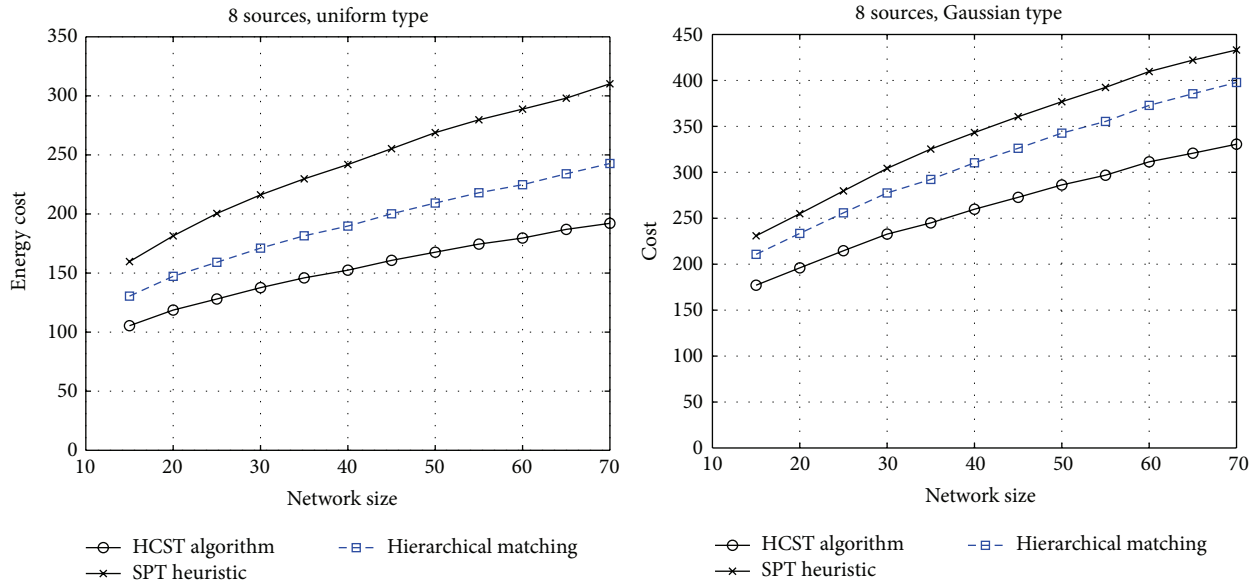
FIGURE 10: An example of randomly generated G for simulation.

FIGURE 11: Energy cost associated with a fixed number of sources.

algorithm again outperforms the SPT heuristic. The relative savings in energy by HCST algorithm ranges 19–41% for uniform type sources and 14–27% for Gaussian type sources. Relative to HM algorithm, HCST algorithm saves energy costs by 20–23% and 14–17% for uniform and Gaussian type sources, respectively. The difference in the energy cost of the algorithms increases with n similar to the case of fixed number of sources, however, such a rate of increase is higher in the case of varying number of sources. This can be explained as follows. When we increase the network size, the number of sources also increases proportionally. When the network size grows, from the previous argument such that there is further room for improvement by HCST, its relative gain will increase with the network size. In addition to

that, since the number of sources grows, the total number of stages at the end of the HCST algorithm will also increase. Since HCST chooses the best tree from the intermediate output trees collected over stages, a large number of stages implies that we can choose the final output tree from a large pool of trees having various degrees of tradeoff between microscopic and macroscopic aspects of the cost reduction. Thus the abundance of source nodes enables us to choose an aggregation tree with a “refined” tradeoff, which is crucial for a convex and decreasing ϕ . This explains the enhanced performance of HCST with increasing number of sources. Hence we conclude from the simulation that the HCST algorithm can improve performance for various proportions of source nodes among the network.

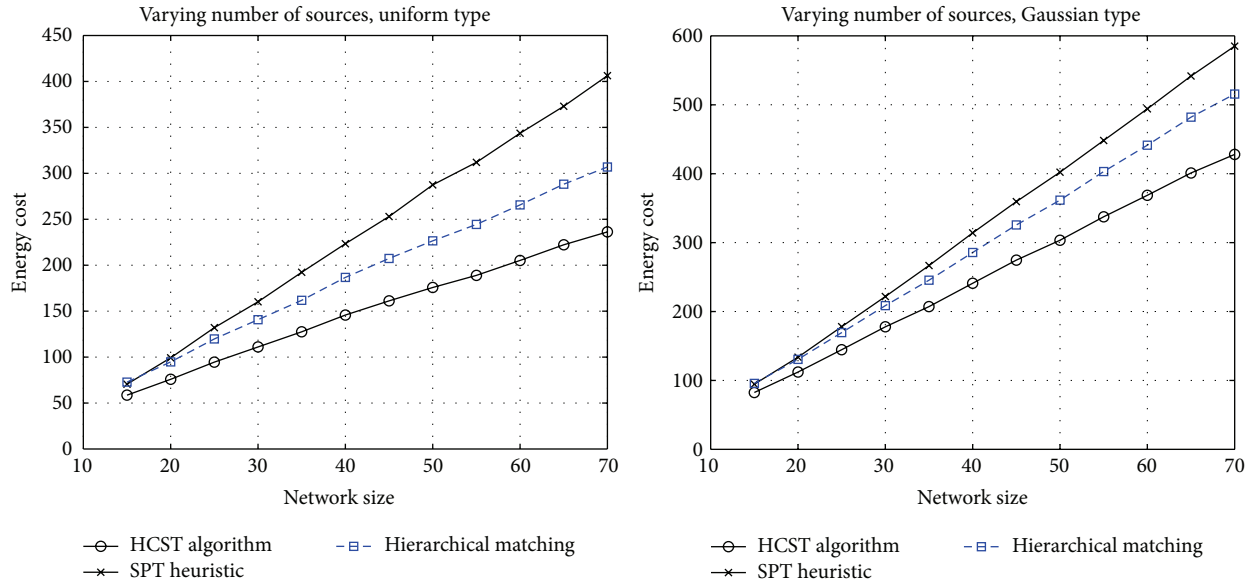


FIGURE 12: Energy cost associated with a varying number of sources.

7. Conclusion

In this paper we have studied a single-sink aggregation problem for wireless sensor networks computing several widely used summary functions. It is observed that the problem is characterized by the aggregation cost function ϕ which maps the amount of aggregated measurements to transmission costs at a link. We show that the properties of ϕ depend heavily on the chosen summary function $g(\cdot)$. When $g(\cdot)$ is given by `sum` or `mean`, we showed that ϕ is concave and increasing, implying that there exist algorithms such as the HM algorithm which can approximate the optimal algorithm by a factor logarithmic in the number of sources. A similar argument was made when $g(\cdot)$ is `weighted sum` for i.i.d. Gaussian sources. When $g(\cdot)$ is given by `max`, however, we have shown that ϕ is convex and decreasing for certain types of sources. For such ϕ we identify that there exists a tradeoff between the following two aspects of cost reduction: firstly local clustering of sources which is the microscopic aspect, and secondly a low-cost routing from the clustered sources to the sink which is the macroscopic aspect. We proposed the Hierarchical Cover and Steiner Tree algorithm which empirically finds the best tradeoff point between the aspects. Numerical examples and simulation results were presented to demonstrate that the HCST algorithm is versatile and improves performance for various types of convex and decreasing ϕ . A future direction would be investigating the optimal aggregation problems for a wider range of summary functions. In addition, the evaluation of the HCST algorithm in a real-world testbed environment is also part of our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by Basic Science Research Program through The National Research Foundation of Korea (NRF) funded by The Ministry of Science, ICT & Future Planning (NRF-2013RIA1A1062500), and in part by the ICT R&D program of MSIP/IITP, (10-911-05-006, High Speed Virtual Router that Supports Dynamic Circuit Network).

References

- [1] A. Giridhar and P. R. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 755–764, 2005.
- [2] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 146–159, 2001.
- [3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 56–67, ACM, August 2000.
- [4] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, pp. 575–578, IEEE, 2002.
- [5] M. Bagaa, Y. Challal, A. Ksentini, A. Derhab, and N. Badache, "Data aggregationscheduling algorithms in wireless sensor networks: solutions and challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1339–1368, 2014.
- [6] L. A. Villas, A. Boukerche, H. S. Ramos, H. A. B. F. de Oliveira, R. B. de Araujo, and A. A. F. Loureiro, "DRINA: a lightweight and reliable routing approach for in-network aggregation in wireless sensor networks," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 676–689, 2013.

- [7] J. Ma, W. Lou, and X.-Y. Li, "Contiguous link scheduling for data aggregation in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1691–1701, 2014.
- [8] Z.-J. Zhang, C.-F. Lai, and H.-C. Chao, "A green data transmission mechanism for wireless multimedia sensor networks using information fusion," *IEEE Wireless Communications*, vol. 21, no. 4, pp. 14–19, 2014.
- [9] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 32–39, 2007.
- [10] P. Von Rickenbach and R. Wattenhofer, "Gathering correlated data in sensor networks," in *Proceedings of the Joint Workshop on Foundations of Mobile Computing*, pp. 60–66, ACM, 2004.
- [11] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked slepianwolf: theory, algorithms, and scaling laws," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4057–4073, 2005.
- [12] S. Patten, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 28–35, Berkeley, Calif, USA, April 2004.
- [13] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2571–2582, March 2004.
- [14] J. Liu, M. Adler, D. Towsley, and C. Zhang, "On optimal communication cost for gathering correlated data through wireless sensor networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 310–321, ACM, September 2006.
- [15] S. Hariharan and N. Shroff, "Maximizing aggregated revenue in sensor networks under deadline constraints," in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 4846–4851, 2010.
- [16] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk," in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pp. 499–505, SIAM, 2003.
- [17] M. Andrews and L. Zhang, "The access network design problem," in *Proceedings of the of IEEE Symposium on Foundations of Computer Science (FOCS '98)*, p. 40, IEEE Computer Society, 1998.
- [18] Y. Bartal, "On approximating arbitrary metrics by tree metrics," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 161–168, ACM, May 1998.
- [19] A. Meyerson, K. Munagala, and S. Plotkin, "Cost-distance: two metric network design," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*, p. 624, IEEE Computer Society, 2000.
- [20] S. Guha, A. Meyerson, and K. Munagala, "A constant factor approximation for the single sink edge installation problems," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 383–388, ACM, 2001.
- [21] M. Andrews, A. Anta, L. Zhang, and W. Zhao, "Routing for energy minimization in the speed scaling model," in *Proceedings of the IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, Calif, USA, March 2010.
- [22] M. Andrews, S. Antonakopoulos, and L. Zhang, "Minimum-cost network design with (Dis)economies of scale," in *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS '10)*, pp. 585–592, Las Vegas, Nev, USA, October 2010.
- [23] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian, "Buy-at-bulk network design: approximating the single-sink edge installation problem," in *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 619–628, Society for Industrial and Applied Mathematics, January 1997.
- [24] B. Awerbuch and Y. Azar, "Buy-at-bulk network design," in *Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science (FOCS '97)*, IEEE Computer Society, pp. 542–547, October 1997.
- [25] D. R. Karger and M. Minkoff, "Building Steiner trees with incomplete global knowledge," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS '00)*, pp. 613–623, IEEE Computer Society, November 2000.
- [26] M. Charikar and A. Karagiozova, "On non-uniform multicommodity buy-at-bulk network design," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pp. 176–182, ACM, 2005.
- [27] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, "Approximation algorithms for non-uniform buy-at-bulk network design," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '06)*, pp. 677–686, Berkeley, Calif, USA, October 2006.
- [28] A. Scaglione and S. D. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," in *Proceedings of the 8th ACM Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, pp. 140–147, Atlanta, Ga, USA, September 2002.
- [29] L. Galluccio, S. Palazzo, and A. T. Campbell, "Efficient data aggregation in wireless sensor networks: an entropy-driven analysis," in *Proceedings of the IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, pp. 1–6, September 2008.
- [30] D. Kandris, P. Tsioumas, A. Tzes, G. Nikolakopoulos, and D. D. Vergados, "Power conservation through energy efficient routing in wireless sensor networks," *Sensors*, vol. 9, no. 9, pp. 7320–7342, 2009.
- [31] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, "Energy efficient and perceived QoS aware video routing over wireless multimedia sensor networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 591–607, 2011.
- [32] L. Galluccio, A. T. Campbell, and S. Palazzo, "Concert: aggregation-based congestion control for sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pp. 274–275, San Diego, California, USA, November 2005.
- [33] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, NY, USA, 1991.
- [34] M. Madiman, "On the entropy of sums," in *Proceedings of the Information Theory Workshop (ITW '08)*, pp. 303–307, IEEE, Porto, Portugal, May 2008.
- [35] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [36] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità, "An improved LP-based approximation for steiner tree," in *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC '10)*, pp. 583–592, June 2010.
- [37] A. Schrijver, *Combinatorial Optimization*, Springer, New York, NY, USA, 2003.