HAL
open science

Optimizing Performance of Ad-hoc Networks Under Energy and Scheduling Constraints

Liron Levin, Michael Segal, Hanan Shpungin

To cite this version:

Liron Levin, Michael Segal, Hanan Shpungin. Optimizing Performance of Ad-hoc Networks Under Energy and Scheduling Constraints. WiOpt'10: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, May 2010, Avignon, France. pp.110-119. inria-00501499

HAL Id: inria-00501499

https://inria.hal.science/inria-00501499

Submitted on 12 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimizing Performance of Ad-hoc Networks Under Energy and Scheduling Constraints

Liron Levin, Michael Segal, *Senior Member, IEEE*, and Hanan Shpungin, *Member, IEEE*

***Abstract*—This paper studies the construction of *power-efficient data gathering tree* for wireless ad hoc networks. Because of their high communication cost and limited capacity, a fundamental requirement in such networks is designing energy efficient data-gathering algorithms to ensure long network survivability. Two possible models for the data gathering problem are explored: *scheduling model* and the *energy model*. In the scheduling model the goal is to minimize the makespan of the most congested node, while in the energy model the goal is to maximize the lifetime of the network. We present a number of provable approximation algorithms and show inapproximation bounds for various versions of data-gathering problem.**

I. Introduction

Wireless ad-hoc networks have found their way to almost every advanced technology in the market; among those are mobile communication, radio broadcasting, and sensor monitoring. The network consists of several transceivers (nodes) located in the plane, communicating by radio. Unlike wired networks, in which the link topology is fixed at the time the network is deployed, wireless ad-hoc networks have no fixed underlying topology. In addition, the relational disposition of wireless nodes is constantly changing. The distribution of the wireless nodes and the different transmission schemes determine the temporary physical topology of the network.

One of the most common and critical tasks for which a wireless ad-hoc network may be deployed is *data gathering* – i.e. each node collects information from its surrounding area and then propagates it, using other nodes as relays, to some base station, also referred to as the *root node*. Many important applications benefit from data gathering scheme, such as habitat monitoring [1], security applications [2] and civil structure monitoring [3]. The information each node collects is encoded into messages, which are then propagated by using a *data gathering tree* ([4], [5], [6]). The tree is a subgraph of the directed communication graph, which represents the underlying physical topology of the network, where nodes correspond to the transceivers and edges correspond to the communication links.

There are two models of propagation: *with* and *without* aggregation. The former model allows each node to accumulate the messages of its descendants and then pass only one fixed size message to its parent in the tree. The latter model requires that *all* messages eventually reach the root node. In this paper we consider data gathering without aggregation, which is a substantially more complicated case than the first, with aggregation, model ([7], [8], [9]).

Efficient construction of the data gathering tree was of interest to the community in previous works ([5], [8], [10]). By efficiency one can relate to many parameters that measure the overall performance of the network. This paper focuses on two different efficiency models – the *scheduling efficiency model* and the *energy efficiency model*.

Scheduling efficiency – The scheduling efficiency model addresses the overall time it takes for all the messages to reach the root node. The time it takes a node to propagate a single message is fixed; thus, the processing time of some node v is proportional to the number of descendants in a subtree rooted at v. As a result, some of the nodes become overloaded and form a bottleneck in the network. We wish to minimize the *makespan* (or completion time) of the most congested node.

Energy efficiency – The energy efficiency model reflects modern-day communication networks, where the nodes are positioned in the Euclidean plane. Each node decides on a transmission power level, and a transmission from node u can be received at node v if the transmission power of u is at least $d(u,v)^\alpha$, where $d(u,v)$ is the Euclidean distance between u and v, where α is a constant representing the *distance-power gradient*, usually taken to be in the interval $[2,4]$ [11]. For simplicity, we assume

This work is supported in part by the Lynn and William Frankel Center for Computer Science and US Air Force European Office of Aerospace Research and Development, grant #FA8655-09-1-3016.

Liron Levin is with Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Michael Segal is with Department of Communication Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Hanan Shpungin is with Department of Computer Science, University of Calgary, Calgary, Canada.

$\alpha = 2$, though our results can be easily extended to any constant power of α. For each node, the amount of energy spent depends on the number of descendants in the data gathering tree and the distance to its parent. The node with the highest energy demand is the first to deplete its energy reserves, which are usually limited and impossible to replenish.

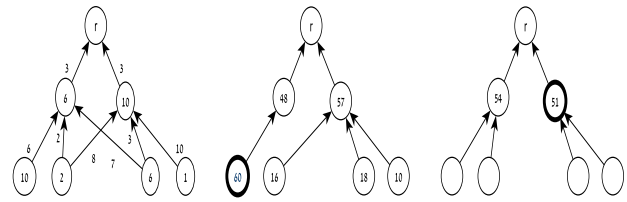
The underlying network topology (the communication graph) has a great impact on the efficiency of the data gathering tree. Although the communication graph is arbitrary (referred to as *general graph* in this paper), certain applications have very specific topology requirements, which we also address: *linear networks* – all nodes are positioned along a straight line, e.g. environmental monitoring along coastlines, undergrounds, rivers and aqueducts [12]); *grid networks* – all nodes are positioned on a grid, e.g. a lake or a bay [13]; *k-layered graphs* – the nodes are partitioned into k tiers, with the the root being in the first tier. A node in tier i can send messages to nodes in tier $i-1$ only. We are especially interested in 3-layered graphs, as this can reduce system complexity and the impact of inefficient random access protocols, avoiding inefficient mobile ad-hoc routing, and alleviating the congestion bottleneck at the Internet gateway model [14]. Also, a 3-layered network coupled with OFDM provides two key benefits: *diversity* and *spatial reuse* gain ([15]).

The paper is organized as follows. In the rest of this section we outline the network model and problem definition, discuss previous work and describe our contribution. Section II addresses the scheduling model, followed by the energy model in Section III.

A. Definition of terms and network model

First, we provide some graph theory related definitions:

- For any directed graph $G = (V, E)$, let E and V denote the edge set and the nodes set, respectively.
- We denote the size of E as m and the size of V as n.
- For any graph G, we define a weight function $\omega : E \to \mathbb{R}$ on the edge set of G and a function $q : V \to \mathbb{Z}^+$ on the node set of G that defines the number of messages each node has.
- We say that a function f on a set X is constant if $f : X \to c$, $c \in \mathbb{Z}^+$.
- For any rooted tree T, we define $T(v)$ to be the subtree rooted at v, including v.
- We define the weight of the tree $o(T(v))$, to be the sum of its nodes weights, i.e. $\sum_{v \in T} q(v)$
- The cost of a node v in T is defined as $C_T(v) = o(T(v)) \cdot w(v, \pi(v))$, where $\pi(v)$ is parent of node v in T.
- Let $\bar{s} = \{s_1, s_2, ..s_k\}$ be the set of children of the root r in T.
- The cost of the root r is zero ($C_T(r) = 0$).



(a) General graph (b) minDG solution (c) maxDG solution

Figure 1: Example for the data gathering problem

Network Model. Consider a wireless network with n nodes and a base station as a directed weighted graph $G = (V, E)$ with a root r. The time is divided into discrete rounds when at every round each node v will sense the environment, gather information and send an arbitrary amount of q messages to the base station. The cost of sending a single message from node v to u is $\omega(v, u)$. We assume that node v can communicate only if it has enough energy to send all q messages in the current round. We define a data gathering tree T to be a reverse arboresence rooted at r. As we pointed above, the communication cost $C_T(v)$ for a node v in T is the cost of sending all the unaggregated messages from all v descendants in T (including that of v) to his father $\pi(v)$. Each node v has an initial unrechargeable battery b_v that drains with every message transmission.

For each model, scheduling and energy, the input graph is defined slightly different. For the scheduling model, denote by G_S the connected directed input graph with n nodes, where each node can communicate with some subset of the nodes in G_S, allowing some fixed processing time per message, i.e. the weight of all the edges in G_S, ω_s is equal to 1. For the energy model, let G_E be a complete directed graph, where sending a message from node u to node v is equal to $\omega_E = d(u, v)^2$– the squared Euclidean distance between the nodes u and v.

Problem Definition. Many important combinatorial problems arise on how to build efficiently data-gathering tree ([8], [10], [5]), where a certain objective function is optimized. We define the general data gathering (DG) problem as follows. The input is a graph $G = (V, E)$, a root r, message quantity q_i for each node v_i, and a function measuring the cost for sending a message from node v_i to v_j, $\omega : E \to \mathbb{R}$. The output is a

convergecast tree $T = (V, E')$ rooted at r that optimizes a given objective function (defined below). Intuitively, given G, we wish to find a data-gathering tree such that the network resources (for example load) are fairly shared between the nodes. If instead of a tree topology we would build a connected spanning tree solution for the data gathering problem, it would force every node to maintain a very large routing table, which makes the obtained structure inapplicable for practical needs. Another practical relaxation that we investigate under both models is the case where each node has only one message to transmit in every round (i.e. $q(v) = 1, \forall v \in V$).

[Minimum Data Gathering problem (minDG)]
Input: Graph G with a weight function $\omega : E \to \mathbb{R}$, a root r and a message quantity function $q : V \to \mathbb{Z}^+$.
Output: A data gathering tree T rooted at r.
Objective: $\min_T(\max_v(C_T(v)))$.

[Maximum Data Gathering problem (maxDG)]
Input: Graph G with a weight function $\omega : E \to \mathbb{R}$, a root r and a message quantity function $q : V \to \mathbb{Z}^+$.
Output: A data gathering tree T rooted at r.
Objective: $\max_T(\min_{v \in \bar{s}}(C_T(v)))$.

Example 1: To illustrate the problems we use the graph in Figure 1a where the numbers represents the initial message size at each node. The final solutions for the general minDG and maxDG problems are shown in Figures 1b and 1c, where the cost of each node v is $C_T(v)$ and is given inside node v.

For the scheduling model the input graph G_S is a connected directed graph with $\omega : E \to 1$. We use the notation **minDGS** and **maxDGS**, respectively, when referring to both problems under this model. We also consider the balanced version of the problem where the objective is $min_T(max_{v \in \bar{s}}(C_T(v)) - min_{v \in \bar{s}}(C_T(v)))$. We call this problem **balancedDGS.** For the energy model G_E is a complete directed graph with a weight function $\omega(u, v) = d(u, v)^2$. We use the notation **minDGE** when referring to the minimum data gathering problem under energy model.

B. Our contribution

In Section II we explore the solution of our problem under the scheduling model. We show $\mathcal{NP}$-hardness proofs, in-approximation results and supplement a number of approximation algorithms for several versions of the problem. Section III is devoted to the energy model, where we show $\mathcal{NP}$-hardness proof and present approximation algorithm for the minDGE problem. In addition, we show approximation solutions for different topologies of underlying input graph such as linear and grid networks. Summary of our results in Figure 2.

C. Previous work

The mathematical foundations of the data gathering problem using the un-aggregated model can br found in ([21]), where Camerini et al. introduced a number of open $\mathcal{NP}$-hard capacitated tree and flow problems (as a matter of fact, most of them have remained open until today). For the un-aggregated model, there are several heuristic solutions. Liang and Liu [10] suggested three heuristics for some variation of minDG problem on the Euclidean plane under the energy model. Their goal was to maximize the network lifetime, and the proposed heuristics are based on a greedy distance tree. Unfortunately, this paper lacks provable achievable performance of the proposed solutions. For the scheduling model, Buragohain et al. [9] proved $\mathcal{NP}$-hardness when the batteries power can vary for each node and proposed a heuristic algorithm and prove that his performance is worse than $\Omega(\frac{\log n}{\log n \log n})$. In fact, it can be shown to be as worse as $\Omega(n)$. Another heuristic solution based on balancing BFS trees can be found at [7]. Additional attempt to solve the un-aggregated problem can be found at [8], [9]. Both papers introduce an integer flow formulation of the problem that leads to some approximate solution (without any guarantee on the algorithm performance). Moreover, the final obtained solution is not a tree. Intensive research has been done for the un-aggregated data gathering problem on fixed topologies. Pan et al. [22] investigated the lifetime problem for the two-tired wireless sensor networks. They provided an approximate solution for 3-layered graph under an energy model that is different for ours. For the grid topology Hui and Han ([13]) supplemented heuristic tree solution. Their greedy heuristic incrementally changes the solution according to the weights of the edges and the number of descendants in each node. A slightly different model where only a subset of the sensors is active in each time unit was suggested by Wang et al. ([23]).

II. Scheduling model

In this section we address the efficient data-gathering problem on different topologies under the scheduling model. We start with a simple optimal solution to those problems on 3-layered graph with uniform message quantity (i.e. the number of messages at each node are the same $q(v) = q(u), \forall u, v \in V$) and show how to approximate the general message quantity case (i.e. $q(v) \in Z^+, \forall v \in V$). Next, we show inapproximation

Approximation Ratio	$\omega(u,v)$	q	f	G	Remarks
2 ([16])	1	$\mathbb{Z}_+$	min max	$k=3$	Inapproximation $\frac{3}{2}$ ([17]), $\mathcal{NP}$-hard for general k (this paper)
$O(\frac{\log\log m}{\log\log\log m})$ ([18])	1	$\mathbb{Z}_+$	max min	$k=3$	Inapproximation 2 ([19]), $\mathcal{NP}$-hard for general k (this paper)
$\log n$ ([20]	1	$\mathbb{Z}_+$	min max	*General*	$\frac{\log n}{3}$ (this paper)
1 (this paper)	1	1	any	$k=3$	
1 (this paper)	$d(u,v)^2$	1	min max	*Complete*	Approximation for linear networks. $\mathcal{NP}$-hard for $q \in \mathbb{Z}^+$and for general graphs (this paper)
$\log n$ (this paper)	$d(u,v)^2$	1	min max	*Complete*	Approximation for grid networks.
$O(\log^2 n)$ (this paper)	$d(u,v)^2$	1	min max	*Complete*	Approximation for for network with uniform distributed nodes.

Figure 2: Summary of the results.

bound for 1-minDGS problem (when each node has only one message to send) and supplement an algorithm that achieves that bound, thus showing that this bound is tight. We start our explanations with some network flow notations that we use through this section.

A $flow\,network$ $N = (G, s, t, c)$ is defined as a directed graph $G = (V, E)$ with a $source$ $s \in V$, a $sink$ $t \in V$ and a *capacity* function $c : E \to \mathbb{R}^+$. A feasible flow f in N is a function $f : E \to \mathbb{R}^+$ that satisfies the following two constraints: $0 \leq f(e) \leq c(e), e \in E$ and $f(in(v)) - f(out(v)) = 0, v \in V - \{s, t\}$, where $in(v)$ is the set of edges entering v, and $out(v)$ is the set of edges leaving v. The goal is to find the maximum flow from s to t. The fastest algorithm for this problem is due to King et al. [24] with running time $O(nm \log_{\frac{m}{n \log n}} n)$. If in addition to constraint (1) we demand a lower bound on the flow in each edge, that is: $l(e) \leq f(e) \leq c(e), e \in E$, then we can use the algorithm from [25] for bounded flows combined with King et al. [24] algorithm for a total running time of $O(nm \log_{\frac{m}{n \log n}} n)$. We also introduce the notation of unsplittable/confluent flow. A flow is said to be *unsplittable/confluent* if the out flow for every node/commodity[1] leaves along a single edge. A directed graph $G = (V, E)$ with a distinct root r is called a **k-layered graph** if all the nodes of the graph can be partitioned into k sets $(X_1, X_2, ..X_k)$ such that a node v that belongs to X_i is connected only to nodes that belong to X_{i-1}, each node has at least one outgoing edge and $X_1 = \{r\}$. Example of a 3-layered graph in shown in Figure 1a.

A. 3-*layered graphs*

We start by showing a polynomial time algorithms that solve the 1-minDGS, 1-maxDGS and 1-balancedDGS problems (in all problems $q(v) = 1, \forall v \in V$). Then we show $\mathcal{NP}$-hardness, inapproximation results and approximation solutions for unrestricted (but polynomially bounded) q.

1) Single message relaxation: First, we create an auxiliary flow network $N = (G, s, t, c)$ by taking a new source node s, connecting s to all the nodes that belongs to X_3 and X_2. We set $c(s, v) = 1, v \in X_2 \cup X_3$, $c(v, w) = 1, v \in X_3, w \in X_2$. The capacity $c \in \mathbb{Z}$ from nodes in X_2 to s will depend on the objective function we use.

Let f_m be the maximum integer flow for this network. Note that if f_c is equal to n then we can relay all the messages from the nodes in the graph to the sink. Also note that setting an upper bound of c on the capacity implies that each node can propagate at most c messages to the sink. Clearly, the lowest possible c for which $f_m = n$ implies an optimal solution to 1-minDGS problem. We can obtain this solution by taking the edges with non zero flow. Therefore, we need to compute $c^* = \{\min c | c \in \{1, 2...n\}\}$ which can be easily obtained by running a binary search over all possible values of c. The total running time of the algorithm is $O(nm \log n \log_{\frac{m}{n \log n}} n)$. In a similar way (by setting lower bounds on the outgoing flow), we can solve the 1-maxDGS problem and the 1-balancedDGS problem.

2) General message quantity relaxation: In this section we show a connection between minDGS problem and maxDGS problem to two well-known combinatorial optimization problems. The first problem is the restricted assignment case of ***Scheduling on Unrelated Parallel Machines*** [17], where we need to schedule n jobs on m machines while minimizing the makespan. That is, we are given a $n \times m$ matrix of non-negative numbers, each entry $p_{i,j}$ denotes the amount of time which machine i needs to process job j and we wish to minimize the maximum processing time. In the restricted assignment version we require that in each row j all the entries are either ∞ or equal to the same value p_j. Lenstra et al. [17] proved that approximating this problem better than 1.5 is $\mathcal{NP}$-hard for the restricted case. They [17] also gave a 2 approximation algorithm for the problem. It is easy to see

[1]The k-commodity flow problem allow a different demand (between s_i and t_i) for each commodity.

that by setting the number of messages at each one of the nodes that belong to X_2 to 1, and by setting the number of messages at each one of the nodes that belong to X_3 to be equal p_j, the formulation of minDGS problem on a 3-layered graph and the stated problem is identical. We can deduce from this result that we cannot approximate minDGS problem with a ratio better than 1.5. We note that if we slightly change the approximation algorithm from [17] we can get a 2-approximation algorithm for our problem.

The second problem is commonly known as ***The Santa Claus problem*** [19]**,** where Santa Claus has a set of n presents that he wants to distribute among a set of m kids and each kid j has value $p_{i,j}$ for each present i. The goal is to distribute the presents in such a way that the least lucky kid is as happy as possible, that is to reach $\max_j \min \sum_i p_{i,j}$. Bezakova et al. [19] proved that this problem cannot be polynomially approximated by a factor better than 2 , where $p_{i,j}$ is equal to p_i or 0, i.e. present i has a constant value for some of the kids and zero value for others. Notice that this problem has the same formulation as maxDGS problem (again setting the number of messages at each node that belongs to X_2 to 1). Thus, maxDGS problem cannot be polynomial approximated better than 2. The best approximation result for the restricted case of the Santa Claus is $O(\frac{\log\log m}{\log\log\log m})$ (see [18]). This approximation algorithm also works for maxDGS problem for 3-layered graph. Note that both minDGS and maxDGS problems are $\mathcal{NP}$-hard on a k-layered graph for every $k > 3$ and solving both problems on general graphs is also $\mathcal{NP}$-hard.

B. General graphs

We first prove that 1-minDGS problem cannot be approximated with a ratio better than $\frac{\log n}{3}$ (implying $\mathcal{NP}$-hardness). A previous attempt to prove the hardness of 1-minDGS problem can be found at [9]. However, their reduction holds only when the initial battery for every node is different. As far as we aware, this is the first $\mathcal{NP}$-hardness proof for 1-minDGS problem even for unit initial battery charge. Finally, we show how to achieve $O(\log n)$ approximation by a polynomial time algorithm for this problem.

1) Inapproximation for 1*-minDGS problem:* We argue that 1-minDGS problem cannot be approximated by a ratio better than $\frac{\log n}{3}$, unless $\mathcal{P} = \mathcal{NP}$. We will use a) the gadget that was first applied by Guruswami et al. ([26]) for the edge-disjoint path problem and b) the underlying tree structure connecting gadgets that was introduced in [20]. First, we need to define the 2-vertex disjoint path problem (which is known to be $\mathcal{NP}$-hard [27]).

2-vertex disjoint directed paths problem:
Instance: A directed graph G with four special nodes s_1, s_2, t_1, t_2.
Problem: Does G contains 2-vertex-disjoint directed paths between $s_1 \rightsquigarrow t_1$ and $s_2 \rightsquigarrow t_2$.
Given an instance $\langle G, s_1, s_2, t_1, t_2\rangle$ of 2-vertex disjoint directed paths problem we create a complete binary tree T with $\log n$ levels ($n \geq V_G$, where V_G is the number of nodes in G). Notice that each level in T contain twice as much nodes as the previous level. We use T to create an auxiliary graph $\tilde{G}$ by transforming every node in T to a copy of G (the original graph). We denote G^i and G^i_v as the ith copies of G, and a copy of a node v in G^i, respectively.

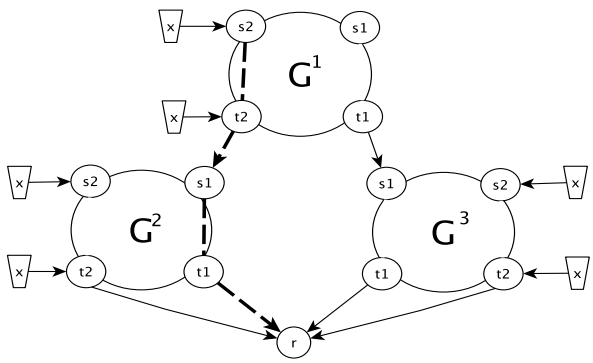
We connect the nodes from level l ($l \geq 1$) of form $G^i_{t_1}$ and $G^i_{t_2}$ ($2^{l-1} \leq i < 2^l$) to nodes in level $l+1$ of form $G^{2i}_{s_1}$ and $G^{2i+1}_{s_1}$, respectively. We attach a directed path with $n \log n$ nodes to each node with the form $G^i_{t_2}$, $G^i_{s_2}$ Finally, we create a sink node r and connect each node that has the form $G^i_{t_1}$, $G^i_{t_2}$ and located in level $\log n$ to r ($2^{\log n-1} \leq i < 2^{\log n}$-1). An example of the construction is shown in Figures 3a and 3b. In Figure 3a we have $n = 4$ with 3 copies (x denotes directed path having $n \log n$ nodes). The connections between the levels are $G^1_{t_1} \rightarrow G^2_{s_1}$ and $G^1_{t_2} \rightarrow G^3_{s_1}$, and $G^2_{t_1}, G^2_{t_2}, G^3_{t_1}, G^3_{t_2}$ are connected to the sink. Denote the cost of the optimal solution to 1-minDGS problem on graph $\tilde{G}$ as OPT. We state the following:

Lemma 1: If G contains 2 vertex disjoint directed paths then $OPT < 3n \log n$.

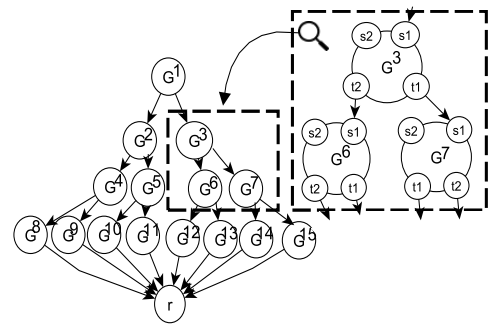
Proof: First note that any path to the sink that starts from any node with form $G^i_{s_1}$ will have at most $n \log n$ nodes (adding at most n nodes to every $G^i_{t_1}$ in each level). Also note that we can create a directed path from every node $G^i_{s_2}$ to the sink with at most $3n \log n$ nodes (the path from $G^i_{s_2}$ to $G^i_{t_2}$ will have most $2n \log n + g$ nodes; then we have the edge from $G^i_{t_2}$ to $G^{2i+1}_{s_1}$ with the following path from $G^{2i+1}_{s_1}$ to the sink resulting in the maximum number of nodes that is equal to $(\log n - 1)g$. This path $G^1_{s_2} \rightsquigarrow G^1_{t_2} \rightarrow G^3_{s_1} \rightsquigarrow G^3_{t_1} \rightarrow r$ in shown by bold dashed lines at Figure 3a. Thus, if G contains 2 vertex disjoint directed paths then $OPT < 3n \log n$, since $n \geq V_G$. ■

Lemma 2: If G does not contains 2 vertex disjoint directed paths then $OPT > n \log^2 n$.

Proof: We define the cost of a node as the sum of nodes that have paths that end in this node (i.e. all the descendants of the node in the optimal solution). We first prove that if the maximum cost of a node in level i is $c \cdot n \log n$ (for any constant c) and G does not contain 2 -vertex disjoint directed paths, then there exists a node in level $i+1$ having cost at least $(c+1) \cdot n \log n$. We



(a) Auxiliary graph G for the case when $n = 4$



(b) Underlying Tree structure T

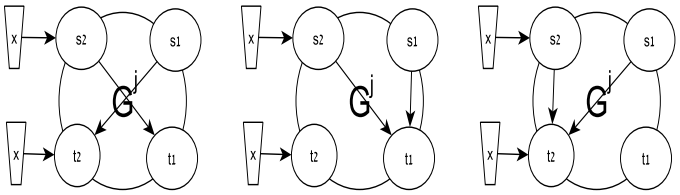
Figure 3: The auxiliary graph $\tilde{G}$

prove this using induction on the number of levels. The base case is obvious (in level 1 to cost of $G^1_{s_2}$ is $n \log n$). For the induction step, suppose there exists a node G^l_k in level i with cost $c \cdot n \log n$. Then one of the nodes $G^j_{s_1}$ in level $i+1$ will cost at least $c \cdot n \log n$, since the path that ends at node G^l_k can only increase its length continuing to level $i+1$. We are facing cases depicted in Figure 4. Note that in each case either $G^j_{t_1}$ or $G^j_{t_2}$ will cost at least $(c+1) \cdot n \log n$. This implies that the induction hypothesis holds and since we have $\log n$ levels, the cost of OPT is at least $n \log^2 n$. ■

Theorem 1: 1-minDGS problem cannot be polynomially approximated within a factor better than $\frac{\log n}{3}$, unless $\mathcal{P} = \mathcal{NP}$.

Proof: Lemma 1 implies that if G contains 2 vertex disjoint directed paths then $OPT < 3n \log n$. Lemma 2 implies that if G does not contain 2 vertex disjoint directed paths, then $OPT > n \log^2 n$. We note that the gap between the two instances is $\frac{n \log^2 n}{3n \log n} = \frac{\log n}{3}$. Now, suppose there exists a polynomial algorithm $\mathcal{A}$ that approximate minDGS with approximation ratio better than $\frac{\log n}{3}$, then if G contains 2-vertex disjoint paths $\mathcal{A}$ will produce a solution of cost at most $\frac{\log n}{3} \cdot 3n \log n = n \log^2 n$. Otherwise, the solution $\mathcal{A}$ produced has a cost of at least $n \log^2 n$ (since $OPT > n \log^2 n$). Consequently, by executing $\mathcal{A}$ we can decide if G contains 2-vertex disjoint sets or not. Thus, unless $\mathcal{P} = \mathcal{NP}$ minDGS cannot be approximated by a ratio better than $\frac{\log n}{3}$. ■

2) A $\log n$ approximation for the minDGS problem: Chen et al. [20] presented an approximation algorithm CONFLT that solves the minimum single commodity flow problem. The input of the algorithm is a graph G, set of sinks s, set of demands d, and a splittable flow f. The output of the algorithm is an un-splittable flow $\widetilde{f}$ having



(a) case (i) (b) case (ii) (c) case (iii)

Figure 4: No 2-disjoint paths

tree topology where the maximum outgoing flow for each node is minimized. CONFLT algorithm guarantees that flow conservation constraints hold in $\widetilde{f}$. It also guarantees that the outgoing flow from each node will leave along a single edge, and if the maximum outgoing flow f in G is 1, then the maximum outgoing flow in $\widetilde{f}$ is $1 + \log n$. For step (2) of BALANCE we use King et al. [24] algorithm for maximum splittable flow, (with running time $O(nm \log_{\frac{m}{n \log n}} n)$. Again, denote by OPT the optimal value to the minDGS problem and by Q the maximum message quantity for all the nodes in the graph. See Algorithm 1 for the implementation.

Lemma 3: By using step (2) of Algorithm BALANCE we can find a flow $\widetilde{f}$ such that the out-flow from any node v will satisfy $\widetilde{f}(v) \leq OPT$.

Proof: The optimal solution to the splittable flow problem can be polynomially solved by any maximum flow algorithm. This solution sets a lower bound on any unsplittable flow that satisfies the demands. ■

Complexity analysis: Running step (2) using King et al. flow algorithm [24] takes $O(\log Qn\, nm \log_{\frac{m}{n \log n}} n)$ time and CONFLT subroutine [20] takes $O(m(m+mn \log \frac{n^2}{m}))$ time. Resulting in a $O(mn(\log Qn \log_{\frac{m}{n \log n}} n + \frac{m}{n} +$

Algorithm 1 BALANCE(G, E, q, r)

1) Create a flow network embedded on G where each node v has a demand $q(v)$. Set the sink to be r.
2) Run a binary search in range $[1..Q]$ to find a flow f and $c^* = \{\min c | c \in \{1, 2...U\}\}$ such that $f(s_i) \leq c^*$ and all the demands from the nodes are satisfied.
3) Normalize the demand of each node (by dividing the original demand by c^*).
4) The set $\bar{s}$ will represent children of r. Remove r from G, obtaining number of sinks.
5) Output CONFLT($G \setminus \{r\}, \bar{s}, \frac{q}{c^*}, f$), where $\frac{q}{c^*}$ is the vector of normalized demands for all nodes.

$\log \frac{n^2}{m}$)) running time algorithm.

III. ENERGY MODEL

In this section we investigate the energy model, where the nodes are deployed in the Euclidean plane and the cost of sending a message is d^2 (where d is the Euclidean distance between the nodes). First, we prove that the minDGE problem is $\mathcal{NP}$-hard in the strong sense[2]. Next, we show how to solve the problem on two special topologies. In the first topology, the nodes are placed on the bi-directional line, and in the second topology they are placed on the unit grid. We show how to achieve the optimal solution for line instance and a $\log n$ approximation solution for the grid topology. Finally, we show how to combine the obtained results to approximate the minDGE problem when the nodes are uniformly distributed in the Euclidean plane.

3) $\mathcal{NP}$-hardness of the minDGE problem: In this section we prove that minDGE problem is $\mathcal{NP}$-hard in the strong sense showing the reduction from the extended version of 3-partition problem that is known to be $\mathcal{NP}$-hard. A previous attempt to prove $\mathcal{NP}$-hardness of this problem can be found at [10]. However, the reduction proposed in that paper is not possible (the node placement scheme in the Euclidean plane is not feasible).

We first introduce the decision version of minDGS problem and 3-partition problem.

Decision version of the minDGE problem:
Instance: A complete graph G where each node has a weight, location, and a cost P.
Question: Is there a solution to minDGE problem with cost less than or equal to P?

***Extended version of* 3*-partition* [28]:**
Instance: A multi-set S of $3m$ elements $S = \{a_1, a_2, .., a_{3m}\}, a_i \in \mathbb{Z}^+$, where the sum of the elements is $m \cdot B$ and the weight of each element is strictly between $\frac{B}{4}$ to $\frac{B}{2}$.
Question: Is there a partition of S into m equal weight sets such that the union of the sets cover S (each set will contain exactly 3-elements)?

Let $I = \langle S \rangle$ be an instance of the extended 3-partition problem. We build a reduction to an instance $\langle G, B+1 \rangle$ of the minDGE problem by defining the following mapping of S to a graph G that serves as the input graph for the minDGE problem. First, for every element a_i we create a node v_i with the weight of a_i, i.e. node v_i has a_i messages to transmit. We position those nodes in $(0, 2.2)$. Next, we create m intermediate nodes $s_1, s_2, ..s_m$ with weight 1 and position them in $(0, 1)$. Those nodes correspond to m bins. We place the root of the graph at coordinates $(0, 0)$.

Lemma 4: minDGE is $\mathcal{NP}$-hard.

Proof: Suppose there exists a solution to minDGE problem on the underlying graph with cost less than or equal to $B + 1$. In this solution: (1) no node v_j from layer-3 deliver a direct message to the root (since sending a direct message from a node in layer-3 cost at least $(2.2)^2 \frac{B}{4} > 1.1B > B + 1$) (2) the node with the maximum cost in the solution to minDGE is an intermediate node (since relaying any message to the intermediate node costs at most $(1.2)^2 \frac{B}{2} < B + 1$). (3) Each intermediate node must relay messages from exactly 3 nodes from layer-3 with total number of messages B, this flow from the pigeonhole principal since the total number of messages in layer-3 is mB and they are divided between m intermediate nodes. Hence, if there is a solution with cost less than or equal to B to minDGE we could find a solution to the extended 3-partition problem by joining each triplet a_i, a_j, a_k with total message weight B that uses intermediate node s_l as a carrier. The opposite direction is clear too. Hence, unless $\mathcal{P} = \mathcal{NP}$ minDGE is $\mathcal{NP}$-hard. ■

4) Optimal solution for the line topology: There are many important applications where the nodes are deployed on a straight line with the sink node (base station) positioned arbitrarily between them. As an example take a railway track monitoring system where sensor nodes with vibrational energy harvesters are placed uniformly along the track to detect wear-and-tear and breakages [12]. We argue that the optimal solution to 1-minDGE problem for the line topology on n equally placed nodes is a directed line (chain) from the fringe nodes to the root. In order to prove this claim, we notice the following observation:

Observation 1: To solve 1-minDGE problem on a straight line we only need to examine the solution when

[2]A problem is said to be $\mathcal{NP}$-hard in the strong sense if it remains $\mathcal{NP}$-hard even when all its numerical parameters are bounded by a polynomial in the length of the input [28].

nodes are positioned only on the positive axis (i.e. if the node are located on $(v_{-n}, ..v_{-1}, r, v_1, ...v_n)$ using only r and $v_1..v_n$.

Theorem 2: The solution to 1-minDGE problem for a line topology with equal distances between the adjacent nodes is a directed chain.

Proof: Using the previous fact we assume that the nodes are ordered from node v_n at location n to the root r at location 0. First observe that the cost of the chain solution is n. Let v_j be the rightmost neighbor of v_1 that deliver a direct message to r, notice that it must be that $j \leq \sqrt{n}$ (otherwise the cost of this node will be at least $j^2 > n$). This implies that the maximum number of messages that node v_j can pass is $\left\lfloor \frac{n}{j^2} \right\rfloor$. Also observe that if node v_j passes a message directly to the root then node v_1 can only deliver $\sum_{i=2}^{n} \left\lfloor \frac{n}{i^2} \right\rfloor + j - 1$ messages to r ($\sum_{i=2}^{n} \left\lfloor \frac{n}{i^2} \right\rfloor$ represent the total number of messages that bypass node v_j and $j-1$ represent the left neighbors of v_j). The total number of messages that can be delivered to r this way are: $\sum_{i=2}^{\sqrt{n}} \left\lfloor \frac{n}{i^2} \right\rfloor + \frac{n}{j^2} + j - 1 < \sum_{i=2}^{\infty} \frac{n}{i^2} + \frac{n}{j^2} + j = 1 - \frac{\pi^2 n}{6} + \frac{n}{j^2} + \sqrt{n}$. For n large enough we get that this is less than $n(0.65 + 0.25 + 0.09) < n$. Thus, n messages can not be delivered to the root. ■

5) Approximating the 1-minDGE problem on the grid topology: The input for the problem is a complete graph with n nodes located on the $\sqrt{n} \times \sqrt{n}$ grid, where the root r is located in the bottom left corner of the grid at coordinates $(0, 0)$, and each node has one message to transmit. The solution for the data gathering problem is a reverse arboresence T having r as a root. We show that on this topology the cost of any reverse arboresence rooted at r is at least $\frac{n}{2 \log n}$. Then we present a deterministic construction of tree T having cost $\frac{n}{2}$, implying $\log n$ approximation solution for this problem.

Lemma 5: The cost of any reverse arboresence rooted at r is at least $\frac{n}{2 \log n}$.

Proof: The total number of messages that must be delivered to r is n. Denote by $d_{i,j}$ the Euclidean distance from a node located at coordinate (i, j) of the grid to r, and by $d_{i,j}^2$ the cost of sending a direct message to r. Assume we have a solution to 1-minDGE problem on the grid with cost p^*. Thus, every node located at (i, j) can relay only $\frac{p^*}{d_{i,j}^2}$ messages to the root. Hence, the total number of messages that can be delivered to r keeping the cost below p^* is: $\sum_{i,j:\sqrt{i^2+j^2}<\sqrt{p^*}} \frac{p^*}{d_{i,j}^2} < \sum_{i=1}^{\sqrt{p^*}} (2i+1) \frac{p^*}{d_{1,i}^2} = \sum_{i=1}^{\sqrt{p^*}} (2i+1) \frac{p^*}{d_{1,i}^2}$. This is equal to: $p^*(H_n(\sqrt{p^*}) + H_{n,2}(\sqrt{p^*})) < p^* 2 \log n$. Since we must deliver n messages to r, p^* is at least $\frac{n}{2 \log n}$. ■

Theorem 3: There is a simple solution with cost $\frac{n}{2} - \sqrt{n}$.

Proof: Passing a directed line through the diagonal nodes directly to the root while moving all the other nodes through the side nodes yields a solution with maximum cost of $\frac{n}{2} - \sqrt{n}$. Thus, this algorithm is a $\log n$ approximation for the problem. ■

6) Approximation algorithm for uniformly distributed nodes in the plane: We use the result from Theorem 3 to achieve an $O(\log^2 n)$ approximation algorithm for the 1-minDGE problem when the nodes are uniformly distributed in the unit square. We refer the reader to [29] for a technical discussion on what is the better topology to model a sensor network: grid or random. For both cases we present provable approximation solutions.

Lemma 6: For n nodes uniformly distributed in a unit square U, if we divide U to a $\sqrt{\frac{n}{\log n}} \times \sqrt{\frac{n}{\log n}}$ grid with equal size cells then w.h.p. (with high probability) we would have at least 1 node in each cell[30] .

Lemma 7: For n nodes uniformly distributed in a unit square U, if we divide U to a $\sqrt{\frac{n}{\log n}} \times \sqrt{\frac{n}{\log n}}$ grid with equal size cells then w.h.p. we would have at most $e^3 \log n$ nodes in each cell, i.e. the load at each cell is at most $e^3 \log n$.

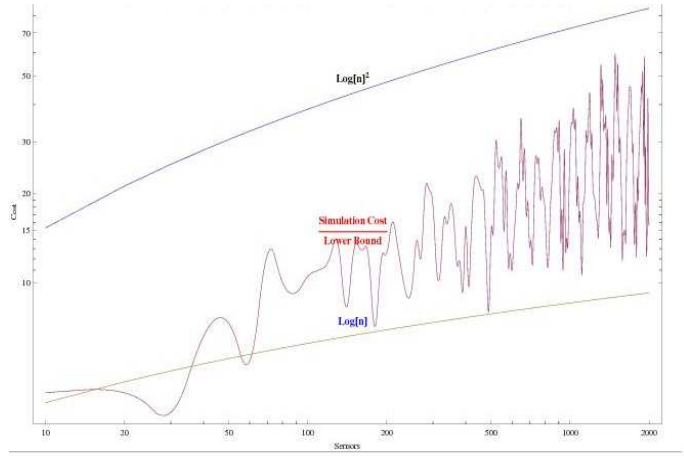
Proof: Assume we have n sensors and $m = \frac{n}{\log n}$ cells. Let $\mathcal{L}$ represent the maximum number of sensor at each cell, from the union bound theory we know that the probability that a specific cell contains at least k sensors is: $\binom{n}{k} \left(\frac{1}{m}\right)^k < \left(\frac{n \cdot e}{k}\right)^k \left(\frac{1}{m}\right)^k$. The probability that any cell will contains at least k sensors is: $m \binom{n}{k} \left(\frac{1}{m}\right)^k < m(\frac{n \cdot e}{k})^k \left(\frac{1}{m}\right)^k$. Setting $m = \frac{n}{\log n}$ and $k = e^3 \log n$, we get that the probability is less than: $\frac{n}{\log n} (\frac{n \cdot e}{e^3 \log n})^{e^3 \log n} (\frac{\log n}{n})^{e^3 \log n} = \frac{1}{n}$. Therefore, $\lim_{n \to \infty} Pr[\mathcal{L} > k] = 0$ as $n \to \infty$ and the lemma holds. ■

Lemma 8: For n nodes uniformly distributed in a unit square U, w.h.p. there is a $O(\log^2(n))$ approximation algorithm for the 1-minDGE problem.

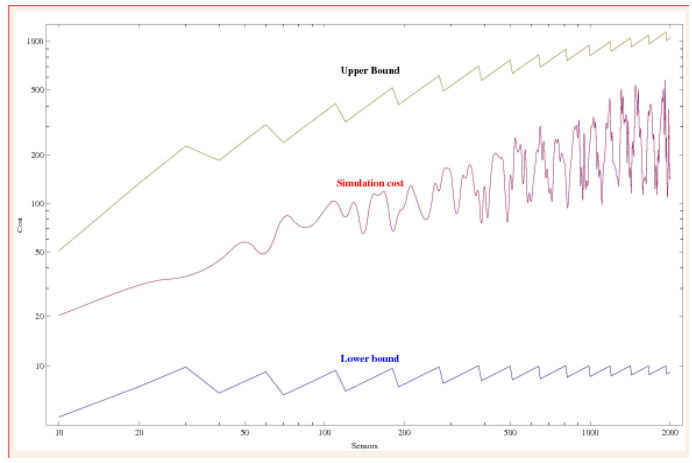
Proof: From the previous section we know that for the unit static grid topology we have a $\log n$ approximation algorithm. Also, from Lemma 6 and Lemma 7 we obtain that if the nodes are uniformly distributed over the unit square we get a grid with $\sqrt{\frac{n}{\log n}} \times \sqrt{\frac{n}{\log n}}$ cells having at least 1 node and at most $e^3 \log n$ nodes in each cell. We deduce that by selecting a cluster leader for each cell, and constructing on top of those leaders the topology suggested in Theorem 3 we achieve an $O(\log^2 n)$ approximation algorithm for the problem. ■

IV. Simulation Study

The main goal of the simulation is to compare the theoretical bounds obtained in the previous section to the experimental performance of the algorithm for 1-minDGE problem with uniformly distributed nodes. Our experimental studies reinforce our claim for a $O(\log^2 n)$ approximation for this problem. All the simulation results are obtained using Mathematica. In our simulation we randomly scatter N nodes in a square region R with $\sqrt{\frac{N}{\log N}} \times \sqrt{\frac{N}{\log N}}$ cells. The positions of the nodes are independently and uniformly distributed in the square region. For every cell we randomly picked a cell leader and constructed a tree topology over the cell leaders, identical to the one in we used in Lemma 3. The reported results are averaged over 30 simulation runs. In Figure 5a we present the ratio between the value of the lower bound of OPT and our simulation results in the range of the expected $[O(\log n)..O(\log^2 n)]$ difference. In Figure 5b we present the cost of the solution to 1-minDGE problem using our simulation program, theoretical upper bound proved at Lemma8 and theoretical lower bound of OPT (Lemma 5).



(a) Ratio of approximation for planar DGE problem



(b) Comparison between theoretical bounds and simulation results

Figure 5: Summary of results

V. Conclusion and Future Work

In this paper the data-gathering problem under different models has been studied. We have investigated the combinatorial nurture of its variants and supplement a number of approximation and inapproximation results for the different relaxations. It would be interesting to close the gap (currently $\log n$) between the lower and upper bounds for the general problem in the plane under energy model and to find a better approximation in the plane for general size messages.

Acknowledgments We thank Refael Hassin for his valuable suggestions during the work on this paper.

References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 88–97.

[2] A. Chehri, P. Fortier, and P.-M. Tardif, "Security monitoring using wireless sensor networks," in *CNSR 2007*. IEEE Computer Society, 2007, pp. 13–17.

[3] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26–34, 2006.

[4] J. Gehrke and S. Madden, "Query processing in sensor networks," *IEEE Pervasive Computing*, vol. 3, no. 1, pp. 46–55, 2004.

[5] Y. Wu, S. Fahmy, and N. B. Shroff, "On the construction of a maximum-lifetime data gathering tree in sensor networks: Np-completeness and approximation algorithm." in *INFOCOM 2008, The 27th Conference On Computer Communication*. IEEE, 2008, pp. 356–360.

[6] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 14–27.

[7] T. Chen, H. Tsai, and C. Chu, "Gathering-load-balanced tree protocol for wireless sensor networks," in *SUTC '06*. IEEE Computer Society, 2006, pp. 8–13.

[8] K. Kalpakis, K. Dasgupta, and P. Namjoshi, "Maximum lifetime data gathering and aggregation in wireless sensor networks," in *ICN'02*, 2002, pp. 685–696.

[9] C. Buragohain, D. Agrawal, and S. Suri, "Power aware routing for sensor databases," in *INFOCOM 2005*. IEEE, 2005, pp. 1747–1757.

[10] W. Liang and Y. Liu, "Online data gathering for maximizing network lifetime in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 2–11, 2007.

[11] K. Pahlavan and A. H. Levesque, "Wireless information networks," 1995.

[12] Z. A. Eu, H. P. Tan, and W. K. Seah, "Routing and relay

node placement in wireless sensor networks powered by ambient energy harvesting," accepted for publication in IEEE WCNC, April 2009.

[13] R. Hui and Han., "A node-centric load balancing algorithm for," in *Proceedings of the IEEE Global Communications Conference*, vol. 1, 2003, pp. 548–552.

[14] Hung-yu Wei and R. D. Gitlin, "Two-hop-relay architecture for next generation wwan/wlan integration," *IEEE Wireless Communications (Special Issue on 4G Mobile Communications - Towards Open Wireless Architecture),*, vol. 11, pp. 24–30, April 2004.

[15] K. Sundaresan and S. Rangarajan, "On exploiting diversity and spatial reuse in relay-enabled wireless networks," in *MobiHoc '08*. ACM, 2008, pp. 13–22.

[16] M. Gairing, B. Monien, and A. Woclaw, "A faster combinatorial approximation algorithm for scheduling unrelated parallel machines," *Theor. Comput. Sci.*, vol. 380, no. 1-2, pp. 87–99, 2007.

[17] J. K. Lenstra, D. B. Shmoys, and É. Tardos, "Approximation algorithms for scheduling unrelated parallel machines," *Math. Program.*, vol. 46, no. 3, pp. 259–271, 1990.

[18] N. Bansal and M. Sviridenko, "The santa claus problem," in *STOC '06*. ACM, 2006, pp. 31–40.

[19] I. Bezáková and V. Dani, "Allocating indivisible goods," *SIGecom Exch.*, vol. 5, no. 3, pp. 11–18, 2005.

[20] J. Chen, R. D. Kleinberg, L. Lovasz, R. Rajaraman, R. Sundaram, and A. Vetta, "(almost) tight bounds and existence theorems for confluent flows," *Theor. Comput. Sci.*, 2004.

[21] P. M. Camerini, G. Galbiati, and F. Maffioli, "The complexity of weighted multi-constrained spanning tree," in *Proc. Colloq. On Theory of Alg.*, 1984, pp. 53–101.

[22] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology control for wireless sensor networks," in *MobiCom '03*. ACM, 2003, pp. 286–299.

[23] X. Wang, Q. Zhang, W. Sun, W. Wang, and B. Shi, "A coverage-based maximum lifetime data gathering algorithm in sensor networks," in *MDM '06*. IEEE Computer Society, 2006, p. 33.

[24] V. King, S. Rao, and R. Tarjan, "A faster deterministic maximum flow algorithm," in *SODA '92*. Society for Industrial and Applied Mathematics, 1992, pp. 157–164.

[25] J. A. Storer, *An Introduction to Data Structures and Algorithms*. Birkhauser Boston, 2001, p. 316.

[26] V. Guruswami, S. Khanna, R. Rajaraman, B. Shepherd, and M. Yannakakis, "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems," *J. Comput. Syst. Sci.*, vol. 67, no. 3, pp. 473–496, 2003.

[27] Fortune, S. Hopcroft, J. E., Wyllie, and J. C., "The directed subgraph homeomorphism problem," *Theoret. Comput. Sci.*, vol. 10, pp. 111–121, 1980.

[28] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[29] S. Panichpapiboon, G. Ferrari, and O. K. Tonguz, "Sensor network with random versus uniform topology : Mac and interference," in *International Confrence on Communication Software and Networks*. IEEE Computer Society, 2009, pp. 604–606.

[30] H. Shpungin and M. Segal, "Near optimal multicriteria spanner constructions in wireless ad-hoc networks," in *INFOCOM 2009*. IEEE, 2009, pp. 163–171.