Finding Facilities Fast *

Saurav Pandit   Sriram V. Pemmaraju

May 6, 2009

Abstract

Clustering can play a critical role in increasing the performance and lifetime of wireless networks. The *facility location* problem is a general abstraction of the clustering problem and this paper presents the first constant-factor approximation algorithm for the facility location problem on unit disk graphs (UDGs), a commonly used model for wireless networks. In this version of the problem, connection costs are not *metric*, i.e., they do not satisfy the triangle inequality, because connecting to a non-neighbor costs ∞. In non-metric settings the best approximation algorithms guarantee an $O(\log n)$-factor approximation, but we are able to use structural properties of UDGs to obtain a constant-factor approximation. Our approach combines ideas from the primal-dual algorithm for facility location due to Jain and Vazirani (*JACM*, 2001) with recent results on the *weighted minimum dominating set* problem for UDGs (Huang et al., *J. Comb. Opt.*, 2008). We then show that the facility location problem on UDGs is inherently *local* and one can solve local subproblems independently and combine the solutions in a simple way to obtain a good solution to the overall problem. This leads to a distributed version of our algorithm in the $\mathcal{LOCAL}$ model that runs in *constant* rounds and still yields a constant-factor approximation. Even if the UDG is specified without geometry, we are able to combine recent results on *maximal independent sets* and *clique partitioning* of UDGs, to obtain an $O(\log n)$-approximation that runs in $O(\log^* n)$ rounds.

1 Introduction

The widespread use of wireless multi-hop networks such as ad hoc and sensor networks pose numerous algorithmic challenges. One of these algorithmic challenges is posed by the need for efficient *clustering* algorithms. Clustering can play a critical role in increasing the performance and lifetime of wireless networks and has been proposed as a way to improve MAC layer protocols (e.g., [11, 31]), higher level routing protocols (e.g., [28, 29, 30]), and energy saving protocols (e.g., [6, 16]). Clustering problems can be modeled as combinatorial or geometric optimization problems of various kinds; the *minimum dominating set (*`MDS`*)* problem, the *k-median* problem, etc. are some popular abstractions of the clustering problem. Since wireless networks reside in physical space and since transmission ranges of nodes can be modeled as geometric objects (e.g., disks, spheres, fat objects, etc.), wireless networks can be modeled as geometric graphs, especially as intersection graphs of geometric objects. This has motivated researchers to consider a variety of clustering problems for geometric graphs [26, 3, 4, 1, 7] and attempt to develop efficient distributed algorithms for these. Most of these clustering problems are NP-hard even for fairly simple geometric graphs and this has motivated attempts to design fast distributed *approximation* algorithms.

In this paper, we present the first constant-factor approximation algorithm for the *facility location* problem on *unit disk graphs (UDGs)*. For points u and v in Euclidean space we use $|uv|$ to denote the Euclidean distance in L_2 norm between u and v. A graph $G = (V, E)$ is a *unit disk graph (UDG)* if there is an embedding of the vertices of G in $\mathbb{R}^2$ (the 2-dimensional Euclidean space) such that $\{u, v\} \in E$ iff $|uv| \leq 1$. The *facility location problem on UDGs* (in short, `UDG-FacLoc`) takes as input a UDG $G = (V, E)$, *opening costs* $f : V \rightarrow \mathbb{R}^+$ associated with vertices, and *connection costs* $c : E \rightarrow \mathbb{R}^+$ associated with the edges. The problem is to find a subset $I \subseteq V$ of vertices to open (as "facilities") and a function $\phi : V \rightarrow I$ that assigns every vertex ("client") to an open facility *in its neighborhood* in such a way that the total cost of opening the facilities and connecting clients to open facilities is minimized. In other words, the problem seeks to minimize the objective function $\sum_{i \in I} f(i) + \sum_{j \in V} c(j, \phi(j))$. See Figure 1 for an illustration. We assume

*Department of Computer Science, The University of Iowa, Iowa City, IA 52242-1419. E-mail: [spandit,sriram]@cs.uiowa.edu. Even though we take blame for the bad title, we confess that the attempted alliteration was inspired by the title of a recent paper: "Leveraging Linial's Locality Limit" by Lenzen and Wattenhofer, to appear in DISC 2008.

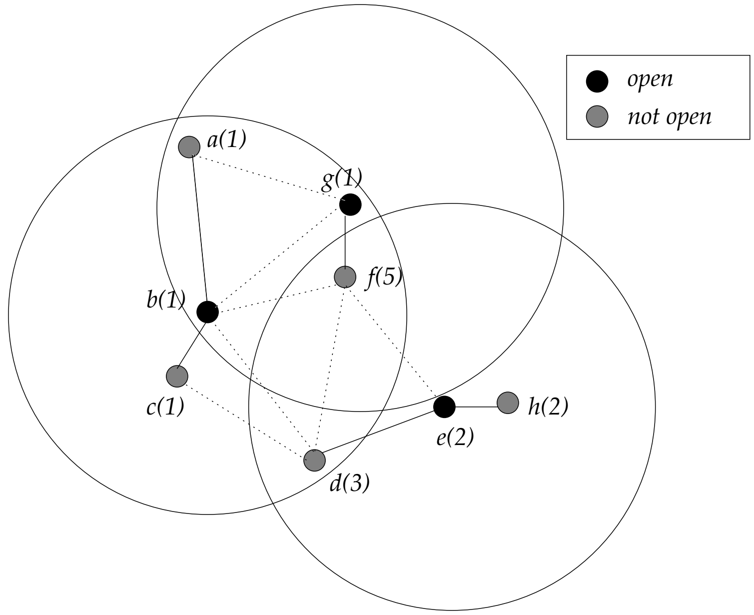


Fig. 1 — A UDG with eight vertices. Opening costs are integers shown next to the vertex names and connection costs of edges are assumed to be Euclidean lengths. Vertices b, g, and e have been opened as facilities. The solid lines indicate the assignments of vertices (clients) to open facilites and the dotted lines indicate edges in the UDG that are not being used for any facility-client connection. Only the disks around the three open facilities are shown in the figure. The cost of this solution is 4 units (for opening facilities) plus $|fg|+|ab|+|cb|+|de|+|he|$.

that the connection costs of edges are determined by their Euclidean lengths via a fairly general function. More precisely, let $g : [0, 1] \to \mathbb{R}^+$ be a monotonically increasing function with *bounded growth*, i.e., for some constant $B \geq 1$, $g(x) \leq B \cdot g(x/3)$ for all $x \in [0, 1]$. We assume that each edge $\{i, j\} \in E$ get assigned a connection cost $c(i, j) = g(|ij|)$. Note that the restriction that g has bounded growth still permits cost functions that are quite general from the point of view of wireless networks. For example, if $g(x) = \beta \cdot x^\gamma$ for constants β and γ (as might be the case if connection costs represent power usage), then $B = 3^\gamma$. It should be noted that every vertex in G is a "client" and every vertex has the potential to be a "facility." Furthermore, a vertex ("client") can only be connected to (i.e., "serviced" by) another vertex ("facility") in its neighborhood and thus the set of open facilities forms a dominating set.

Note that `UDG-FacLoc` is inherently *non-metric*, i.e., connection costs of edges do not satisfy the triangle inequality. This is because a vertex cannot be connected to a non-neighbor, implying that the connection cost of a vertex to a non-neighbor is ∞. There are no known constant-factor approximation algorithms for the non-metric version of facility location, even for UDGs. In one sense, this is not surprising because `UDG-FacLoc` is a generalization of the *weighted minimum dominating set (WMDS)* problem on UDGs. This can be seen by noting that an instance of `WMDS`, namely $G = (V, E)$, $w : V \to \mathbb{R}^+$, can be interpreted as a `UDG-FacLoc` instance in which the connection costs (of edges) are set to 0 and each opening cost $f(i)$ is set to the vertex weight $w(i)$. There have been no constant-factor approximation algorithms for `WMDS` on UDGs until recently, with the result of Ambühl et al. [1] being the first constant-factor approximation for `WMDS` on UDGs. Subsequently, Huang et al. [13] have improved the approximation ratio significantly. Our technique combines the well known *primal-dual* algorithm of Jain and Vazirani [15] with these recent constant-factor approximation algorithms for `WMDS` on UDGs, to obtain a constant-factor approximation for `UDG-FacLoc`. Applicability of our technique to more general models of wireless networks, for example, *unit ball graphs* in higher dimensional spaces or *doubling metric spaces*, *disk graphs*, *growth-bounded graphs*, etc. is only limited by the availability of good approximation algorithms for the `WMDS` problem on these graph classes. Using our technique, a constant-factor approximation algorithm for `WMDS` on any of these graph classes would immediately imply a constant-factor approximation for facility location on that graph class.

UDGs are simple and popular models of wireless networks and the facility location problem on UDGs is a general abstraction of the clustering problem on wireless networks. For more background see the recent survey by Frank [8] on the facility location problem as it arises in the context of wireless and sensor networks.

Unlike the WMDS problem that ignores the cost of connecting to *dominators*, the facility location problem explicitly models *connection costs*. As a result, solutions to WMDS may lead to clustering that is quite poor (e.g. refer to Figure 2). To be more specific consider one common application of dominating sets in wireless

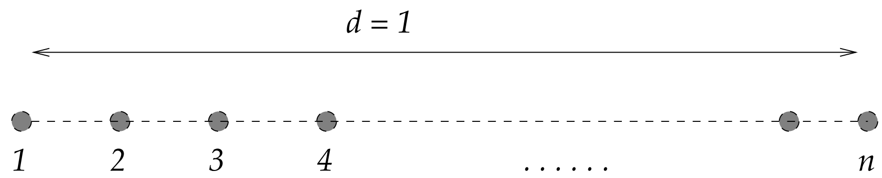
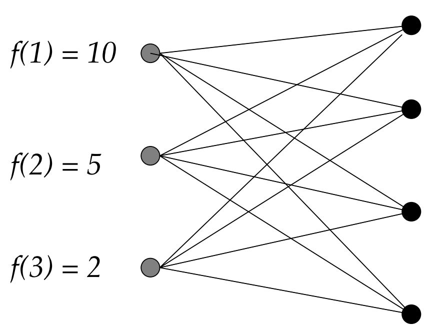


Fig. 2 — An instance of UDG-FacLoc where n vertices are uniformly spaced on a line segment of unit length. Suppose that for all i, $f(i) = 1/n$. Opening all vertices as facilities would cost a total of 1 unit, with connection costs being 0. An optimal WMDS solution would include just one vertex at cost $1/n$. If connection costs are assumed to be Euclidean distances, then the "connection cost" of the WMDS solution would be $\Theta(n)$.

networks, which is to save energy by sending all dominatees into a low power *sleep mode* and having the network be serviced exclusively by the dominators. While it makes sense to keep the size or weight of the dominating set small so that most nodes are in the sleep mode, ignoring the connection costs could yield a dominating set in which each dominator has to spend a lot of energy in order to reach its dominatees. By using an objective function that takes opening costs as well as connection costs into account, UDG-FacLoc yields a set of cluster heads that can service the network with smaller overall cost and for a longer duration.

1.1 Related work

Facility location is an old and well studied problem in operations research ([18, 25, 2, 17, 5]), that arises in contexts such as locating hospitals in a city or locating distribution centers in a region. A *standard* instance of the facility location problem takes as input a complete bipartite graph $G = (F, C, E)$, where F is the set of facilities and C is the set of cities, opening costs $f : F \rightarrow \mathbb{R}^+$, and connection costs $c : E \rightarrow \mathbb{R}^+$. The goal, as mentioned before, is to find a set of facilities $I \subseteq F$ to open and a function $\phi : C \rightarrow I$ that assigns every city to an open facility so as to minimize $\sum_{i \in I} f(i) + \sum_{j \in C} c(j, \phi(j))$. See Figure 3 for an illustration. In



	1	2	3
a	4	1	6
b	7	1	9
c	5	1	7
d	4	6	1

c(i,x) values

Fig. 3 — An instance of standard FacLoc. The table shows the pairwise connection costs between clients and facilities. OPT consists of open facilities 2 and 3 with clients a, b and c connected to facility 2 and client d to facility 3. Total cost of OPT is 11. Note that any solution with a single open facility or with all the facilites open, will have cost more than 11. So is the case for any solution that opens facility 1.

this context, the connection costs are said to satisfy the *triangle inequality* if for any $i, i' \in F$ and $j, j' \in C$, $c(i,j) \leq c(i,j') + c(i',j') + c(i',j)$. In the *metric facility location* problem the connection costs satisfy the triangle inequality; when they don't we have the more general *non-metric facility location* problem. UDG-FacLoc can be seen as an instance of the non-metric facility location problem by setting $F = V$, $C = V$, setting connection costs between a facility and a city that correspond to non-adjacent vertices to

∞, setting $c(i,i) = 0$ for all $i \in V$, and inheriting the remaining connection costs and opening costs in the natural way. $O(\log n)$-approximation algorithms for the non-metric facility location problem are well known [12, 19]. Starting with an algorithm due to Shmoys, Tardos and Aardal [24] the metric facility location problem has had a series of constant-factor approximation algorithms, each improving on the approximation factor of the previous. In this paper we make use of an elegant primal-dual schema algorithm due to Jain and Vazirani [15] that provides a 3-approximation to the metric facility location problem. Since `UDG-FacLoc` is not a metric version of the facility location problem, we cannot use the Jain-Vazirani algorithm directly. We use the Jain-Vazirani algorithm to get a "low cost," but infeasible solution to `UDG-FacLoc` and then "repair" this solution via the use of a "low weight" dominating set and show that the resulting solution is within a constant-factor of `OPT`.

Several researchers have attempted to devise distributed algorithms for the facility location problem; these attempts differ in the restrictions placed on the facility location problem and in the network and distributed computing models. For example, Moscibroda and Wattenhofer [20] present a distributed algorithm for the standard non-metric facility location problem. The network on which their algorithm runs is the complete bipartite graph on F, the set of facilities and C, the set of cities. Since this network has diameter 2, one way to solve the problem would be for a node to gather information about the entire network in constant number of communication rounds and just run a known sequential algorithm locally. Thus this problem is uninteresting in the $\mathcal{LOCAL}$ model [21] of distributed computation. The problem becomes interesting in the $\mathcal{CONGEST}$ model, where a reasonable bound, such as $O(\log n)$ bits, is imposed on each message size. In such a model, exchanging a lot of information costs a lot of rounds and Moscibroda and Wattenhofer [20] present an approximation algorithm for non-metric facility location that, for every k, achieves an $O(\sqrt{k}(m\rho)^{1/\sqrt{k}}\log(m+n))$-approximation in $O(k)$ communication rounds. Here m is the number of facilities, n is the number of clients, and ρ is a coefficient that depends on the numbers (i.e., opening costs and connection costs) that are part of the input. The main thrust of this result is that even with a constant number of communication rounds, a non-trivial approximation factor can be achieved. However, it should be noted that no matter how large k is (e.g., $k = \text{polylog}(n)$), the approximation factor of this algorithm is $\Omega(\log(m+n))$.

Frank and Römer [9] consider facility location on multi-hop networks (like we do), but assume that given edge weights, the connection cost $c(i,j)$ for any pair of vertices i and j is simply the shortest path distance between i and j. This turns their problem into a *metric* problem and thus they can use known sequential algorithms; in particular, they use the 1.61-approximation due to Jain et al. [14]. Frank and Römer [9] show how to implement the sequential algorithm of Jain et al. [14] in a distributed setting without any degradation in the approximation factor, but they do not provide any non-trivial running time guarantees. Frank and Römer [9] do mention the version of the problem in which connection costs between non-neighboring vertices is ∞, but they just observe that since this is a non-metric problem, constant-factor approximation algorithms are not known.

Gehweiler et al. [10] present a constant-approximation, constant-round distributed algorithm using only $O(\log n)$-bits per message, for the *uniform* facility location problem. In this problem, all opening costs are identical and the underlying network is a clique. The authors make critical use of the fact that all facility opening costs are identical in order to obtain the constant-approximation. The uniform opening costs assumption is restrictive for certain settings. For example, if we want opening costs to reflect the amount of battery power that nodes have available – more the available power at a node, cheaper it is to open that node, then this assumption requires the battery power at all nodes to remain identical through the life of the network. This may be untenable because nodes will tend to expend different amounts of power as they perform different activities. The interesting aspect of the Gehweiler et al. [10] algorithm is that all message sizes are bounded above by $O(\log n)$.

1.2 Main results

We assume that we are given a UDG along with its geometric representation. Let $g : [0,1] \rightarrow \mathbb{R}^+$ be a monotonically increasing function with *bounded growth*, i.e., there exists a constant B such that $g(x) \leq B \cdot g(x/3)$ for all $x \in [0,1]$. Each edge $\{i,j\} \in E$ gets assigned a connection cost $c(i,j) = g(|ij|)$, representing the dependence of the connection cost on the Euclidean distance between the involved vertices. For any $\varepsilon > 0$, we present a $(6 + B + \varepsilon)$-approximation algorithm for `UDG-FacLoc`. To put this result in context, observe that if connection costs are exactly Euclidean distances, i.e., $g(x) = x$, then $B = 3$ and we have a $(9 + \varepsilon)$-approximation. If the connection costs are meant to represent energy usage, then a function such as $g(x) = \beta \cdot x^\gamma$ for constants β and $2 \le \gamma \le 4$ may be reasonable. In this case, $B = 3^\gamma$ and we get a $(3^\gamma + 6 + \varepsilon)$-approximation, still a constant-factor approximation. We then present a distributed implementation of our algorithm that runs in just $O(1)$ rounds and yields an $O(B)$-approximation. To obtain this result we show that `UDG-FacLoc` can be solved "locally" with only a constant-factor degradation in the quality of the solution. One aspect of our result, namely the constant approximation factor, depends crucially on the availability of a geometric representation of the input UDG. If we are given only a combinatorial representation of the input n-vertex UDG, then our algorithm runs in $O(\log^* n)$ rounds yielding an $O(\log n)$-approximation. This result depends on two recent results: (i) an $O(\log^* n)$-round algorithm for computing a *maximal independent set (MIS)* in growth-bounded graphs [23] and (ii) an algorithm that partitions a UDG, given without geometry, into relatively small number of cliques [22]. Overall, our results indicate that `UDG-FacLoc` is as "local" a problem as MIS is, provided one is willing to tolerate a constant-factor approximation. Our techniques extend in a straightforward manner to the *connected* `UDG-FacLoc` problem, where it is required that the facilities induce a connected subgraph; we obtain an $O(1)$-round, $O(B)$-approximation for this problem also.

2 Sequential Algorithm

Now we present a high level *three step* description of our algorithm for finding a constant-factor approximation for `UDG-FacLoc`. Let $G = (V, E)$ be the given UDG with an opening cost $f(i)$ for each vertex $i \in V$ and connection cost $c(i, j)$ for each edge $\{i, j\} \in E$. We assume that there is a monotonically increasing function $g : [0, 1] \to \mathbb{R}^+$ satisfying $g(x) \le B \cdot g(x/3)$ for all $x \in [0, 1]$ for some $B \ge 1$, such that $c(i, j) = g(|ij|)$.

Step 1. Convert the given instance of `UDG-FacLoc` into a standard non-metric instance of facility location. This transformation is as described in the previous section. Run the primal-dual algorithm of Jain and Vazirani [15] on this instance to obtain a solution S. The solution S may contain connections that are infeasible for `UDG-FacLoc`; these connections have connection cost ∞ and they connect pairs of non-adjacent vertices in G.

Step 2. Assign to each vertex i of G a weight equal to $f(i)$. Compute a dominating set of G with small weight. For this we can use the $(6 + \varepsilon)$-approximation algorithm due to Huang et al. [13]. Let D^* denote the resulting solution.

Step 3. For each vertex $j \in V$ that is connected to a facility by an edge of cost ∞, reconnect j to an arbitrarily chosen neighbor $d \in D^*$. Think of the vertices $d \in D^*$ as facilities and declare them all open. Let the new solution to `UDG-FacLoc` be called S^*.

We will prove the following theorem in the next subsection.

Theorem 1 *Let OPT denote the cost of an optimal solution to a given instance of `UDG-FacLoc`. Then $cost(S^*) \le (6 + B + \varepsilon) \cdot OPT$.*

2.1 Analysis

To analyze our algorithm we need some details of the Jain-Vazirani primal-dual algorithm used in Step 1. For a more complete description see [15]. The starting point of this algorithm is the following Integer Program (IP) representation of facility location. Here y_i indicates whether facility i is open and x_{ij} indicates if city j is connected to facility i. The first set of constraints ensure that every city is connected to a facility and the second set of constraints guarantee that each city is connected to an open facility.

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{i \in F, j \in C} c(i,j) \cdot x_{ij} + \sum_{i \in F} f(i) \cdot y_i & \\
\text{subject to} & \displaystyle\sum_{i \in F} x_{ij} \ge 1, & j \in C \\
& y_i - x_{ij} \ge 0, & i \in F, j \in C \\
& x_{ij} \in \{0,1\}, & i \in F, j \in C \\
& y_i \in \{0,1\}, & i \in F
\end{array}$$

As is standard, we work with the LP-relaxation of the above IP obtained by replacing the integrality constraints by $x_{ij} \geq 0$ for all $i \in F$ and $j \in C$ and $y_i \geq 0$ for all $i \in F$. The dual of this LP-relaxation is the following:

$$\begin{array}{lll} \text{maximize} & \sum_{j \in C} \alpha_j & \\ \text{subject to} & \alpha_j - \beta_{ij} \leq c(i,j), & i \in F, j \in C \\ & \sum_{j \in C} \beta_{ij} \leq f(i), & i \in F \\ & \alpha_j \geq 0, & j \in C \\ & \beta_{ij} \geq 0, & i \in F, j \in C \end{array}$$

The dual variable α_j can be interpreted as the amount that city j is willing to pay in order to connect to a facility. When $\alpha_j \geq c(i,j)$ for any i, then out of α_j, $c(i,j)$ goes towards paying for connecting to facility i, whereas the "extra," namely β_{ij}, is seen as the contribution of city j towards opening facility i. Initially all the α_j and β_{ij} values are 0. The Jain-Vazirani algorithm initially raises all of the α_j values in synch. When α_j reaches $c(i,j)$ for some edge $\{i,j\}$, then the connection cost $c(i,j)$ has been paid for by j and any subsequent increase in α_j is accompanied by a corresponding increase in β_{ij} so that the first dual constraint is not violated. The quantity β_{ij} is j's contribution towards opening facility i and when there is enough contribution, i.e., $\sum_j \beta_{ij} = f(i)$, then the facility i is declared *temporarily open*. Furthermore, when facility i is temporarily opened, all unconnected cities j that are making positive contribution towards $f(i)$, i.e., $\beta_{ij} > 0$, are declared *connected* to i. Also, any unconnected city j that has completely paid its connection cost $c(i,j)$, but has not yet started paying towards β_{ij}, i.e., $\alpha_j = c(i,j)$ and $\beta_{ij} = 0$, is also declared *connected* to j. The opening of a facility i corresponds to setting $y_i = 1$ and declaring a city j connected to i corresponds to setting $x_{ij} = 1$. Once a facility i is open and cities connected to it, then the dual variables of these cities are no longer raised; otherwise the dual constraint $\sum_{j \in C} \beta_{ij} \leq f(i)$ would be violated. The algorithm proceeds in this way until every city has been connected to some open facility. This is the end of Phase 1 of the algorithm.

It is easy to check that at the end of Phase 1, $\{\alpha_j, \beta_{ij}\}$ define a feasible dual solution and $\{y_i, x_{ij}\}$ define a feasible *integral* solution. If the cost of the primal solution is not too large compared to the cost of the dual solution, then by the Weak Duality Theorem, we would have a solution to facility location that is not too far from a lower bound on OPT. However, the gap between the costs of the dual and the primal solutions can be quite high because a single city may be contributing towards the connection costs and opening costs of many facilities. To fix this problem, Phase 2 of the algorithm is run. Let F_t be the set of temporarily open facilities. Define a graph H on this set of vertices with edges $\{i, i'\}$ whenever there is a city j such that $\beta_{ij} > 0$ and $\beta_{i'j} > 0$; in other words, city j is contributing a positive amount towards the opening of both facilities i and i'. Compute a *maximal independent set (MIS)* I of H and declare all facilites in I open (permanently) and close down all facilities in $F_t \setminus I$, i.e., set $y_i = 0$ for all $i \in F_t \setminus I$. Due to the shutting down of some facilities, some cities may be connected to closed facilities implying that the primal solution may be infeasible, due to violation of the $y_i - x_{ij} \geq 0$ constraints. Call a city j a *Class I city* if it is connected to an open facility. Denote the set of Class I cities by C_1. We'll call cities outside of C_1, *Class II cities*. At this point in the algorithm the primal and the dual solution satisfy the following properties.

Lemma 2 ***[Jain-Vazirani [15]]*** *The dual solution $\{\alpha_j, \beta_{ij}\}$ is feasible. The primal solution $\{y_i, x_{ij}\}$ is integral, but may not be feasible. Furthermore,*

$$\sum_{j \in C_1} \alpha_j = \sum_{j \in C_1} c(j, \phi(j)) + \sum_{i \in I} f(i).$$

The above lemma is essentially saying that the Class I cities completely pay for connections to and the opening of facilities in I. The goal now is to fix the infeasibility of the primal solution, i.e., find connections for cities outside C_1, without increasing the cost of the primal solution too much relative to the cost of the dual. Let j be a city that is connected to a closed facility. If there is a open facility i to which j has already paid connection cost, i.e., $\alpha_j \geq c(i,j)$, then simply connect j to one such city. Since $\alpha_j \geq c(i,j)$, the connection cost is paid for by α_j and furthermore the opening cost of i has been paid for by other cities. This leaves a set C' of cities such that for each $j \in C'$, $\alpha_j < c(i,j)$ for all open facilities i. This may happen,

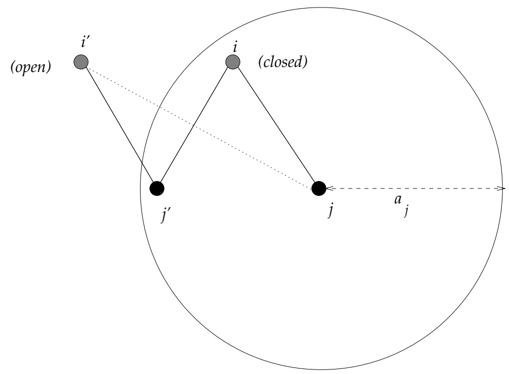


Fig. 4 — Client j is connected to temporarily open facility i at the end of Phase 1. Client j' contributes positively to the opening cost of both i and i'. Facility i is closed at the beginning of Phase 2 and facility i' becomes a candidate for connecting j to.

for example, if none of j's neighbors in G have been opened as facilities and therefore for every open facility i, $c(i, j) = \infty$. Note that at the end of Phase 1, there was a temporarily open facility, say i, to which j was connected and in Phase 2, i was shut down. This implies that (i) $\alpha_j \geq c(i, j)$ and (ii) there exists a city j' that is paying a positive amount towards the opening of two facilities i and i' and this "double payment" is responsible for i being shut down. See Figure 2.1 for an illustration. In such a case, the Jain-Vazirani algorithm simply connects j to i'. In the *metric* facility location case, Jain and Vazirani are able to show that the connection cost $c(j, i')$ is not too big relative to $c(i, j)$ (they show, $c(j, i') \leq 3 \cdot c(i, j)$). In our case, i' may be outside the neighborhood of j and therefore $c(j, i') = \infty$ and therefore connecting j to i' is too costly. This possible mistake is fixed in the subsequent two steps of our algorithm, via the use of a `WMDS` solution. We now include the last two steps of our algorithm in the analysis to show that we are able to find a facility that is not too costly for j to connect to. More precisely, if i' is in the neighborhood of j, then $c(i', j) < \infty$ and we are able to show that connecting j to i' is a good idea. On the other hand, if i' is not a neighbor of j, then we show that connecting j to some neighbor in the `WMDS` solution D^* will not increase the cost of the solution too much. We make use of the following inequalities that Jain and Vazirani prove. The first inequality was mentioned earlier in this paragraph, but the remaining two inequalities take a little bit of work to prove and we refer the reader to the Jain-Vazirani paper [15].

Lemma 3 ***[Jain-Vazirani [15]]*** *$\alpha_j \geq c(i, j)$, $\alpha_j \geq c(i, j')$ and $\alpha_j \geq c(i', j')$.*

Lemma 4 *Let $B \geq 1$ be a constant satisfying $g(x) \leq B \cdot g(x/3)$ for all $x \in [0, 1]$. If i' is a neighbor of j, then j is connected to i' in Step 1 and $c(i', j) \leq B \cdot \alpha_j$. If i' is not a neighbor of j, then j is connected to some neighbor $i^* \in D^*$ in Step 3 and $c(i^*, j) \leq B \cdot \alpha_j$.*

Proof: Since Euclidean distances satisfy triangle inequality, we have

$$|ij| + |ij'| + |i'j'| \geq |i'j|.$$

Let y denote the largest of the three terms on the left hand side above. Then $y \geq |i'j|/3$. Suppose that i' is a neighbor of j. Then $|i'j| \leq 1$ and $c(i', j) = g(|i'j|) < \infty$. Then,

$$\begin{aligned} c(i', j) = g(|i'j|) &\leq B \cdot g\left(\frac{|i'j|}{3}\right) && \text{(due to bounded growth of } g\text{)} \\ &\leq B \cdot g(y) && \text{(due to monotonicity of } g\text{)} \\ &\leq B \cdot \alpha_j && \text{(due to Lemma 3).} \end{aligned}$$

Now suppose that i' is not a neighbor of j. Then $|i'j| > 1$ and for any neighbor i^* of j, $|i'j| > |i^*j|$. Since $y \geq |i'j|/3$, it follows that $y > |i^*j|/3$. Then, by the same reasoning as above, we get

$$\begin{aligned} c(i^*, j) = g(|i^*j|) &\leq B \cdot g\left(\frac{|i^*j|}{3}\right) \\ &\leq B \cdot g(y) \\ &\leq B \cdot \alpha_j. \end{aligned}$$

□

Lemma 5 *Let S^* be the solution produced by our algorithm. Then, $cost(S^*) \leq (6 + B + \varepsilon) \cdot OPT$, where OPT is the cost of an optimal solution to* `UDG-FacLoc`*.*

Proof: The cost of the entire solution can be expressed as

$$\left(\sum_{i \in I} f(i) + \sum_{j \in C_1} c(\phi(j), j)\right) + \left(\sum_{i \in D^*} f(i) + \sum_{j \in C_2} c(\phi(j), j)\right).$$

By Lemma 2, the first term in the above sum equals $\sum_{j \in C_1} \alpha_j$. Let OPT_{DOM} denote the weight of an optimal dominating set when each vertex i of G is assigned weight $f(i)$. Then,

$$\sum_{i \in D^*} f(i) \leq (6 + \varepsilon) \cdot OPT_{DOM} \leq (6 + \varepsilon) \cdot OPT \tag{1}$$

because we use the $(6 + \varepsilon)$-approximation algorithms of Huang et al. [13] to compute a dominating set of small weight. Also, by Lemma 4,

$$\sum_{j \in C_2} c(\phi(j), j) \leq B \cdot \sum_{j \in C_2} \alpha_j. \tag{2}$$

Together the above inequalities yield

$$\begin{aligned} cost(S^*) &= \left(\sum_{i \in I} f(i) + \sum_{j \in C_1} c(\phi(j), j)\right) + \left(\sum_{i \in D^*} f(i) + \sum_{j \in C_2} c(\phi(j), j)\right) \\ &\leq \sum_{j \in C_1} \alpha_j + (6 + \varepsilon) \cdot OPT + B \cdot \sum_{j \in C_2} \alpha_j \\ &\leq B \cdot \sum_{j \in C} \alpha_j + (6 + \varepsilon) \cdot OPT \qquad \text{(since } B \geq 1\text{)} \\ &\leq (B + 6 + \varepsilon) \cdot OPT \qquad \text{(by Weak Duality Theorem).} \end{aligned}$$

□

3 Distributed Algorithm

In this section, we present an $O(1)$-round distributed implementation of the above algorithm in the $\mathcal{LOCAL}$ model [21]. In the $\mathcal{LOCAL}$ model there is no upper bound placed on the message size and due to this, a node can collect all possible information (i.e., node IDs, topology, interactions) about its k-neighborhood in k communication rounds. We show in this section that `UDG-FacLoc` is inherently a "local" problem provided we are willing to tolerate a constant-factor approximation in the cost of the solution. This property of `UDG-FacLoc` allows us to solve a version of the problem independently on *small squares* and combine the solutions in a simple way to get the overall solution. We partition the plane into squares by placing on the plane an infinite grid of $1/\sqrt{2} \times 1/\sqrt{2}$ squares. This is a standard and simple way of partitioning a UDG with geometric representation into cliques. The square S_{ij} for $i, j \in \mathbb{Z}$, contains all the points (x, y) with $\frac{i}{\sqrt{2}} \leq x < \frac{i+1}{\sqrt{2}}$ and $\frac{j}{\sqrt{2}} \leq y < \frac{j+1}{\sqrt{2}}$. Let $G = (V, E)$ be the given UDG. For a square S_{ij} that has at least one node in V, let $V_{ij} \subseteq V$ be the set of vertices that lie in S_{ij}. Let $N(V_{ij})$ denote the set of all vertices in $V \setminus V_{ij}$ that are adjacent to some vertex in V_{ij}. Now consider the subproblem, denoted $\texttt{UDG-FacLoc}_{ij}$, in which we are allowed to open facilities from the set $V_{ij} \cup N(V_{ij})$ with the aim of connecting all the nodes in V_{ij} as clients to these facilities. The objective function of the problem remains the same: minimize the cost of opening facilities plus the connection costs. See Figure 3 for an illustration.

Let $\{F_{ij}, \phi_{ij}\}$ denote a solution to $\texttt{UDG-FacLoc}_{ij}$, where $F_{ij} \subseteq V_{ij} \cup N(V_{ij})$ is the set of open facilities and $\phi_{ij} : V_{ij} \to F_{ij}$ is the assignment of clients to open facilities. Let $\cup_{ij}\{F_{ij}, \phi_{ij}\}$ denote a solution to `UDG-FacLoc` in which the set of open facilities is $\cup_{ij} F_{ij}$ and the assignment $\phi : V \to \cup_{ij} F_{ij}$ is defined by $\phi(v) = \phi_{ij}(v)$ if $v \in V_{ij}$. Thus $\cup_{ij}\{F_{ij}, \phi_{ij}\}$ defines a simple way of combining solutions of $\texttt{UDG-FacLoc}_{ij}$ to obtain a solution of `UDG-FacLoc`. The following lemma shows that if the small square solutions $\cup_{ij}\{F_{ij}, \phi_{ij}\}$

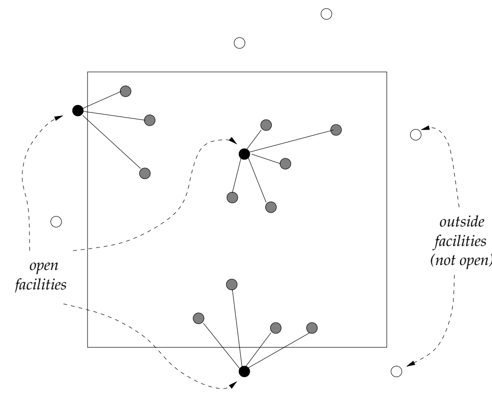


Fig. 5 — An instance and a possible solution of UDG-FacLoc$_{ij}$. Each vertex inside the square has to be connected as a client to an open facility. Any vertex that resides inside the square or outside the square can be opened as a facility as long as a client inside the square can connect to it. Note that the "outside vertices" more than 1 unit away from all the "inside vertices" can be ignored, giving us a local instance of UDG-FacLoc.

are good then combining them in this simple way yields a solution to UDG-FacLoc that is also quite good. This lemma is a generalization of a result due to Ambühl et al. [1] that was proved in the context of the WMDS problem for UDGs.

Lemma 6 *For each $i, j \in \mathbb{Z}$, let OPT_{ij} denote the cost of an optimal solution to UDG-FacLoc$_{ij}$ and let $\{F_{ij}, \phi_{ij}\}$ be a solution to UDG-FacLoc$_{ij}$ such that for some c, $cost(\{F_{ij}, \phi_{ij}\}) \leq c \cdot OPT_{ij}$. Then $cost(\cup_{ij}\{F_{ij}, \phi_{ij}\}) \leq 16c \cdot OPT$. Here OPT is the cost of an optimal solution to UDG-FacLoc.*

Proof: Let $\{F^*, \phi^*\}$ be an optimal solution to UDG-FacLoc, i.e., $cost(F^*, \phi^*) = OPT$. For any $i, j \in \mathbb{Z}$, let $\{F^*(S_{ij}), \phi^*(S_{ij})\}$ denote the restriction of $\{F^*, \phi^*\}$ to square S_{ij}. More precisely, $F^*(S_{ij}) = F^* \cap (V_{ij} \cup N(V_{ij}))$ and $\phi^*(S_{ij})$ is the restriction of ϕ^* to the domain V_{ij}. Note that $\phi^*(S_{ij})$ maps every vertex in V_{ij} to some vertex in $F^*(S_{ij})$ and therefore $\{F^*(S_{ij}), \phi^*(S_{ij})\}$ is a feasible solution to UDG-FacLoc$_{ij}$, implying that

$$OPT_{ij} \leq cost(F^*(S_{ij}), \phi^*(S_{ij})). \tag{3}$$

Furthermore,

$$\begin{aligned}
\sum_{i,j} cost(F^*(S_{ij}), \phi^*(S_{ij})) &\leq 16 \cdot \sum_{i \in F^*} f(i) + \sum_{j \in V} c(j, \phi^*(j)) \\
&\leq 16 \cdot \left(\sum_{i \in F^*} f(i) + \sum_{j \in V} c(j, \phi^*(j)) \right) \\
&= 16 \cdot OPT.
\end{aligned}$$

The first inequality above follows from the fact that a unit disk can intersect at most 16 squares of dimensions $1/\sqrt{2} \times 1/\sqrt{2}$. This implies that a facility in F^* may be servicing clients from at most 16 different squares and therefore may appear at most 16 times in the sum on the left hand side. Finally, we bound the cost of the solution obtain by simply "unioning" the small square solutions as follows:

$$\begin{aligned}
cost(\cup\{F_{ij}, \phi_{ij}\}) &\leq \sum_{i,j} cost(F_{ij}, \phi_{ij}) \\
&\leq c \cdot \sum_{i,j} OPT_{ij} \qquad \text{(by lemma hypothesis)} \\
&\leq c \cdot \sum_{i,j} cost(F^*(S_{ij}), \phi^*(S_{ij})) \qquad \text{(from (3))} \\
&\leq 16c \cdot OPT.
\end{aligned}$$

□

The above lemma implies the following simple distributed algorithm.

Step 1. Each node v gathers information (i.e., coordinates of nodes, opening costs of nodes, and connection costs of edges) about the subgraph induced by its 2-neighborhood.

Step 2. Each node v in S_{ij} then identifies V_{ij} and $N(V_{ij})$. Recall that $V_{ij} \subseteq V$ is the set of nodes that belong to square S_{ij} and $N(V_{ij}) \subseteq V \setminus V_{ij}$ is the set of nodes outside of V_{ij} that have at least one neighbor in V_{ij}.

Step 3. Each node v locally computes the solution of $\texttt{UDG-FacLoc}_{ij}$, thereby determining whether it should be opened as a facility and if not which neighboring facility it should connect to.

Based on the above description, it is easily verified that the algorithm takes 2 rounds of communication. Note that the instance of $\texttt{UDG-FacLoc}_{ij}$ solved in Step 3 is slightly different from `UDG-FacLoc`, in that only certain vertices (namely, the vertices in V_{ij}) need to connect to open facilities, whereas every vertex (both in V_{ij} and in $N(V_{ij})$) is a potential facility. This difference is minor and the $(6+B+\varepsilon)$-approximation algorithm described in the previous section, can be essentially used without any changes, to solve $\texttt{UDG-FacLoc}_{ij}$. Lemma 6 then implies that the distributed algorithm above would yield a $16 \cdot (6 + B + \varepsilon)$-approximation algorithm. We can do better by making use of an intermediate result due to Ambühl et al. [1] that presents a 2-approximation algorithm for the `WMDS` problem on each square S_{ij}. Using arguments from the previous section, we can use this to obtain a $(B+2)$-approximation for $\texttt{UDG-FacLoc}_{ij}$ and a $16\cdot(B+2)$-approximation for `UDG-FacLoc`.

4 Solving the Problem without Geometry

The distributed algorithm described in the above section depends crucially on the given UDG's geometric representation. In this section we sketch a distributed algorithm running in $O(\log^* n)$ rounds that yields an $O(\log n)$-approximation to `UDG-FacLoc`, when the input UDG $G = (V, E)$ is given without any associated geometric information.

Step 1. Compute a maximal independent set (MIS) I of G. Since G is *growth bounded* this can be done in $O(\log^* n)$ rounds using the recent algorithm of Schneider and Wattenhofer [23].

Step 2. Assign each vertex v in $V \setminus I$ to an arbitrary neighbor in I. For each $v \in I$ let S_v denote the set $\{v\} \cup \{u \mid u \text{ is assigned to } v\}$. Partition each S_v into a constant number of cliques using the algorithm due to Pemmaraju and Pirwani [22].

Step 3. Let $C_1, C_2, \ldots, C_t$ be the resulting cliques from the above step. For each i, $1 \leq i \leq t$, define $\texttt{UDGFacLoc}_i$ as the subproblem in which vertices in C_i are the clients and these have to be connected to facilities chosen from $C_i \cup N(C_i)$, where $N(C_i)$ is the set of all vertices that have a neigbor in C_i.

Step 4. Solve $\texttt{UDG-FacLoc}_i$ independently and combine the solutions to the subproblems in the simple way described in the previous subsection.

Two remarks about this algorithm sketch are in order. First, we do not know how to obtain a constant-factor approximation for $\texttt{UDG-FacLoc}_i$ because we do not know how to obtain a constant-factor approximation to `WMDS` in this non-geometric setting. The best we can do is use a simple greedy algorithm to obtain an $O(\log n)$-approximation to `WMDS`. This is what leads to the $O(\log n)$ approximation factor for $\texttt{UDG-FacLoc}_i$. Second, the clique-partitioning algorithm of Pemmaraju and Pirwani [22] yields a cluster graph with degree bounded above by a constant. This means that Lemma 6 holds in the non-geometric setting also (for a constant that is different than 16) and as a result the $O(\log n)$-approximation for $\texttt{UDG-FacLoc}_i$ leads to an $O(\log n)$-approximation for `UDG-FacLoc` as well.

Thus if we could improve the approximation factor of the `WMDS` algorithm to a constant, we would get a constant-factor algorithm for `UDG-FacLoc`, running in $O(\log^* n)$ rounds. It is worth pointing out that the usual greedy algorithm can produce a dominating set whose weight is $\Omega(\log n)$ times the weight of OPT even for `WMDS` instances on small squares. More precisely, consider a vertex-weighted UDG $G = (V, E)$ such that V is partitioned into two sets I and O, where I is the set of vertices that lie inside a $1/\sqrt{2} \times 1/\sqrt{2}$ square and every vertex in O has at least one neighbor in I. The problem is to find $O^* \subseteq O$ of minimum total weight such that O^* is a dominating set for I, i.e., every vertex in I has a neighbor in O^*. The standard greedy algorithm for this problem would repeatedly pick from O a vertex v that maximizes the coverage per unit weight, i.e., ratio of the number of as yet uncovered neighbors in I to $w(v)$. Figure 6 shows a simple instance of this problem for which this greedy algorithm yields a set of vertices whose weight is $\Omega(\log n)$ times the weight of OPT.

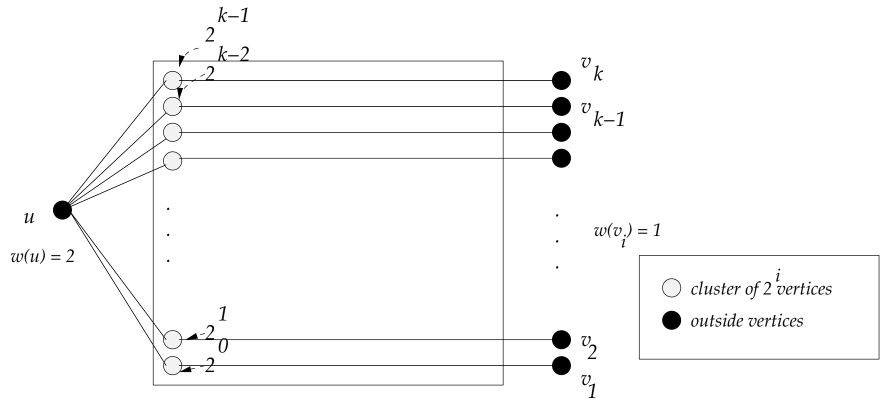


Fig. 6 — In the first step, the greedy algorithm will pick v_k over u because v_k covers 2^{k-1} uncovered vertices per unit weight, whereas u covers $2^{k-1} - \frac{1}{2}$ uncovered vertices per unit weight. In every subsequent step, a v_i will be picked over u and we get a dominating set of weight k, whereas $OPT = 2$. Since $k = \Theta(\log n)$, the lower bound follows.

5 Future Work

One open question implied by this work is whether we can obtain a constant-factor approximation algorithm for facility location on more general classes of wireless network models. We believe that a first step towards solving this problem would be to obtain a constant-factor approximation algorithm for `UDG-FacLoc` when the input UDG is given without any geometry. The only obstacle to obtaining such an approximation, using our techniques, is the lack of a constant-factor approximation to `WMDS` on UDGs given without geometry. We intend to focus on this problem in the future.

The distributed algorithm we present runs in $O(1)$ rounds in the $\mathcal{LOCAL}$ model, which assumes that message sizes are unbounded. Our algorithm depends on each node v being able to gather information about its 2-neighborhood in $O(1)$ rounds. This volume of communication is clearly not possible in $O(1)$ rounds in the $\mathcal{CONGEST}$ model. We would like to extend our distributed algorithm to the $\mathcal{CONGEST}$ model.

Finally, we would like to investigate the effect of non-uniform demands and capacities on the complexity of facility location problems on UDGs.

References

[1] Christoph Ambühl, Thomas Erlebach, Matús Mihalák, and Marc Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *APPROX-RANDOM*, pages 3–14, 2006.

[2] M. L. Balinski. On finding integer solutions to linear programs. In *Proceedings of IBM Scientific Computing Symposium on Combinatorial Problems*, pages 225–248, 1966.

[3] Vittorio Bilò, Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Geometric clustering to minimize the sum of cluster sizes. In *ESA*, pages 460–471, 2005.

[4] Xiuzhen Cheng, Xiao Huang, Deying Li, Weili Wu, and Ding-Zhu Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[5] G. Cornuejols, G. Nemhouser, and L. Wolsey. *Discrete Location Theory.* Wiley, 1990.

[6] Budhaditya Deb and Badri Nath. On the node-scheduling approach to topology control in ad hoc networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 14–26, 2005.

[7] Thomas Erlebach and Erik Jan van Leeuwen. Domination in geometric intersection graphs. *Lecture Notes in Computer Science*, 4957:747–758, 2008.

[8] Christian Frank. *Algorithms for Sensor and Ad Hoc Networks*. Springer, 2007.

[9] Christian Frank and Kay Römer. Distributed facility location algorithms for flexible configuration of wireless sensor networks. In *Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2007)*, pages 124–141, Santa Fe, NM, USA, June 2007.

[10] Joachim Gehweiler, Christiane Lammersen, and Christian Sohler. A distributed o(1)-approximation algorithm for the uniform facility location problem. In *SPAA '06: Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 237–243, New York, NY, USA, 2006. ACM.

[11] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, 2000.

[12] Dorit S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982.

[13] Yaochun Huang, Xiaofeng Gao, Zhao Zhang, and Weili Wu. A better constant-factor approximation for weighted dominating set in unit disk graph. *Journal of Combinatorial Optimization*, 2008.

[14] Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *J. ACM*, 50(6):795–824, 2003.

[15] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[16] Jaewon Kang, Yanyong Zhang, and Badri Nath. Analysis of resource increase and decrease algorithm in wireless sensor networks. In *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 585–590, 2006.

[17] Leon Kaufman, Marc Vanden Eede, and Pierre Hansen. A plant and warehouse location problem. *Operational Research Quarterly*, 28(3):547–554, 1977.

[18] Alfred A. Kuehn and Michael J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):643–666, 1963.

[19] Jyh-Han Lin and Jeffrey Scott Vitter. e-approximations with minimum packing constraint violation (extended abstract). In *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 771–782, 1992.

[20] Thomas Moscibroda and Roger Wattenhofer. Facility location: distributed approximation. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 108–117, 2005.

[21] David Peleg. *Distributed computing: a locality-sensitive approach*. Society for Industrial and Applied Mathematics, 2000.

[22] Sriram Pemmaraju and Imran Pirwani. Good quality virtual realizations of unit ball graphs. In *Algorithms - ESA 2007*, pages 311–322, 2007.

[23] Johannes Schneider and Roger Wattenhofer. A Log-Star Distributed Maximal Independent Set Algorithm for growth-Bounded Graphs. In *27th ACM Symposium on Principles of Distributed Computing (PODC), Toronto, Canada*, August 2008.

[24] David B. Shmoys, Éva Tardos, and Karen Aardal. Approximation algorithms for facility location problems (extended abstract). In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 265–274, 1997.

[25] John F. Stollsteimer. A working model for plant numbers and locations. *Management Science*, 45(3):631–645, 1963.

[26] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 281–290, 2004.

[27] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[28] Peng-Jun Wan, Khaled M. Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.*, 9(2):141–149, 2004.

[29] Yu Wang and Xiang-Yang Li. Localized construction of bounded degree and planar spanner for wireless ad hoc networks. In *DIALM-POMC '03: Proceedings of the 2003 joint workshop on Foundations of mobile computing*, pages 59–68, 2003.

[30] Yu Wang, Weizhao Wang, and Xiang-Yang Li. Distributed low-cost backbone formation for wireless ad hoc networks. In *MobiHoc*, pages 2–13, 2005.

[31] Tao Wu and Subir Biswas. Minimizing inter-cluster interference by self-reorganizing mac allocation in sensor networks. *Wireless Networks*, 13(5):691–703, 2007.