Distributed Approximation of Cellular Coverage

Boaz Patt-Shamir[1,⋆], Dror Rawitz[1], and Gabriel Scalosub[2]

[1] School of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel,
{boaz,rawitz}@eng.tau.ac.il
[2] Department of Computer Science, University of Toronto, Toronto, ON, Canada,
scalosub@cs.toronto.edu

Abstract. We consider the following model of cellular networks. Each base station has a given finite capacity, and each client has some demand and profit. A client can be covered by a specific subset of the base stations, and its profit is obtained only if its demand is provided in full. The goal is to assign clients to base stations, so that the overall profit is maximized subject to base station capacity constraints.
In this work we present a distributed algorithm for the problem, that runs in polylogarithmic time, and guarantees an approximation ratio close to the best known ratio achievable by a centralized algorithm.

1 Introduction

In future cellular networks, base stations capacities, as well as clients diversity, will become a major issue in determining client coverage and service. The main service provided by current cellular networks is voice traffic, which has relatively small bandwidth requirement, compared to the capacity available at the base stations. However, in future 4G cellular networks the services offered by cellular providers are expected to require higher rates, and client diversity is expected to increase. Such services include video traffic, and other high-rate data traffic. In such settings, maximizing the usage of available resources would become a more challenging task, and as recent evidence has shown, current solutions might end up being far from optimal.

In this work we address one of the basic optimization problems arising in such settings, namely, the assignment of clients to base stations, commonly known as *cell selection*. We take into account both base stations diversity, encompassed by (possibly) different capacities for each base station, as well as clients diversity, encompassed by different clients having different demands, and different profits. A major obstacle in tackling this problem is due to the fact that, naturally, different base stations have different coverage areas, and therefore can contribute to covering only some subset of the set of clients.

The currently used scheme for assigning clients to base stations is the greedy best-SNR-first approach, where clients and base stations interact locally, and assignment is made in a greedy and local manner. This approach might provide reasonable performance when clients demands are small with respect to the base stations capacity (as is

⋆ Supported in part by the Israel Science Foundation, grant 664/05.

the case for voice traffic), and all clients are considered equally profitable. However, as clients' demands increase, the resource utilization degrades considerably. This degradation was the main concern of a recent work by Amzallag et al. [1], who propose several global mechanisms for determining the assignment of clients to base stations, while providing guarantees as to their performance. However, these mechanisms are based on a centralized approach: effectively, it is assumed that information of the entire network is gathered by a central server, which locally finds an assignment of clients, and then distributed back to the base stations. This approach suffers from the usual drawbacks of a centralized approach, such as inability to scale to large numbers.

In this paper we present an efficient *distributed* algorithm that computes an assignment. In our algorithm, clients and base stations communicate locally, and after a polylogarithmic number of rounds of communication agree upon an assignment, without resorting to a centralized algorithm with global knowledge. Our algorithm is robust and handles both various base stations capacities, as well as clients' heterogeneous demands, and variable profits. We give worst-case guarantees on the performance of our algorithm (with high probability), and show its approximation ratio is arbitrarily close to the best known ratio of centralized solution. To state our results more precisely, let us first formalize the problem and the computational model.

1.1 Problem Statement and Model

We consider the following model. An instance of the *Cellular Coverage problem* (CC) consists of the following components.

- A bipartite graph $G = (I, J, E)$ where $I = \{1, 2, \dots, m\}$ is a set of *base stations* and $J = \{1, 2, \dots, n\}$ is a set of clients. An edge (i, j) represents the fact that client j can receive service from base station j.
- A *capacity* function c that maps each base station i to a non-negative integer $c(i)$ called the capacity of i.
- A *demand* function d that maps each client j to a non-negative integer $d(j)$ called the demand of j.
- A *profit* function p that maps each client j to a non-negative integer $p(j)$ called the profit of j.

The output of CC is a partial assignment of clients to base stations, where a client may be assigned only to one of its neighboring base stations, and such that the total demand of clients assigned to a base station does not exceed its capacity. The goal is to maximize the sum of profits of assigned clients.

Given a constant $r \leq 1$, an instance of CC is said to be r*-restricted* if for every $(i, j) \in E$, we have $d(j) \leq r \cdot c(i)$, i.e., no client demands more than an r-fraction of the capacity of a base station from which it may receive service. The r-CC problem is the CC problem where all instances are r-restricted.

Let (I, J, E, c, d, p) be an instance of CC. For every base station i we let $N(i) \subseteq J$ denote the set of clients which can be covered by i, and for every client j we let $N(j) \subseteq I$ denote the set of base stations which can potentially cover j. For any set of clients or base stations A, we let $N(A) = \bigcup_{v \in A} N(v)$. Given any function f (e.g., the demand, profit, or capacity), we let $f(A) = \sum_{v \in A} f(v)$. Given any subset of clients $S \subseteq J$, we define $\overline{S} = J \setminus S$.

Let x be a mapping that assigns clients to base stations, and let $\mathrm{clients}(x)$ denote the set of clients assigned by x to some base station. For a base station $i \in I$ let $\mathrm{load}_x(i) = \sum_{j\,:\,x(j)=i} d(j)$, i.e., $\mathrm{load}_x(i)$ is the sum of all demands assigned to i. We further let $\mathrm{res}_x(i)$ denote the residual capacity of i, i.e., $\mathrm{res}_x(i) = c(i) - \mathrm{load}_x(i)$. For a client $j \in J \setminus \mathrm{clients}(x)$ and a base station $i \in N(j)$ we say that j *is eligible for* i if $\mathrm{res}_x(i) \geq d(j)$.

Model of Computation. We consider the standard synchronous message passing distributed model of computation (cf. the CONGEST model of [2]). Briefly, the system is modeled as an undirected graph, where nodes represent processing entities and edges represent communication links. Execution proceeds in synchronous rounds, each round consists of three substeps: first, each node may send a message over each of its incident links; then nodes receive all messages sent to them in that round; and finally some local computation is carried out. The length of every message is restricted to $O(\log n)$ bits, where n is the number of nodes in the system. Nodes may have unique identifiers of $O(\log n)$ bits. Note that in our model, the communication graph is identical to the input graph of CC.

1.2 Our Results

We present a distributed randomized algorithm for the r-CC problem, which, given any $\gamma \in (0, 1]$, runs in time $O(\gamma^{-2} \log^3 n)$, and guarantees, with high probability, to produce an assignment with overall profit at least a $\frac{1-r}{2-r}(1-\gamma)$ fraction of the optimal profit possible.

We note that our running time is affected by the best running time of a distributed algorithm finding a maximal matching, which we use as a black box. The best algorithm up to date, due to [3], has expected running time $O(\log n)$, which is reflected by one of the logarithmic factors of our running time. Any improvement in such an algorithm to time $O(T)$ would immediately imply a running time of $O(\gamma^{-2} T \log^2 n)$ for our algorithm.

1.3 Previous Work

There has been extensive work done on cell selection, and client assignment to base stations in the networking community, focusing on aspects of channel allocation, power control, handoff protocols, and cell site selection (e.g., [4–7]).

Our proposed model was studied in the offline setting in [1]. They study two types of setting, where the first allows the coverage of a client by at most one base station, and the other allows covering a client by more than one base station, whereas its profit is obtained only if its entire demand is satisfied. They refer to the former as the *cover-by-one* paradigm, and to the latter as the *cover-by-many* paradigm. They present a local ratio algorithm for the r-CC problem using the cover-by-many paradigm which is guaranteed to produce a $(1-r)$-approximate solution. This algorithm is based upon a simpler algorithm using the cover-by-one paradigm which is guaranteed to produce a $\frac{1-r}{2-r}$-approximate solution, and this is with respect to the best possible cover-by-many solution.

The CC problem using the cover-by-one paradigm is also closely related to the multiple knapsack problem with assignment constraints, for which a special case where clients demands equal their profits is considered in [8]. They present several approximation algorithms for this problem, starting from a randomized LP-rounding algorithm which produces a $\frac{1}{2}$-approximate solution, through an algorithm employing a sequential use of the FPTAS for solving a single knapsack problem which produces a $(\frac{1}{2}-\varepsilon)$-approximate solution, and finally a greedy algorithm which is guaranteed to produce a $\frac{1}{3}$-approximate solution. Another related problem is the general assignment problem considered in the offline settings in [9–11] (see also references therein).

Although the offline problem, and its various variants, has received considerable amount of attention in recent years, we are not aware of any attempts to solve any of the above problems in a distributive manner. Some very restricted cases of the problem, namely, where all the capacities and all the demands are the same, can be viewed as matching problems, and hence distributed algorithms for solving them are available (see [12] for an overview of these results). Specifically, for the subcase where clients profits are arbitrary, the problem reduces to finding a maximum weight matching in the underlying bipartite graph, whereas in the subcase where all clients profits are the same, the goal is to find a maximum cardinality matching. For both these problems the best solutions are due to [12]. For the former problem they show how to obtain a $(\frac{1}{2}-\varepsilon)$-approximate solution, whereas for the latter problem, they guarantee a $(1-\varepsilon)$-approximate solution. Another closely related problem is that of finding a maximal matching in an unweighted graph for which there exists a distributed algorithm [3]. All of the above algorithms are randomized, and their expected running time is logarithmic in the number of nodes.

Paper Organization. In Section 2 we discuss the centralized version of the problem. In Section 3 we present the details of our distributed algorithm, and analyze its performance guarantee and running time. In Section 4 we present some extensions of our results, and finally, in Section 5 we conclude and discuss some open questions.

2 A Centralized Approach

In this section we explain a centralized algorithm for CC which we later implement in a distributed model. The idea is to use the *local ratio* approach [13, 14], which in our case boils down to an extremely simple greedy algorithm: compute, for each client, its profit-to-demand ratio; scan clients in decreasing order of this ratio, and for each client in turn, assign it to a base station if possible, or discard it and continue. However, to facilitate the analysis, we present this algorithm in recursive form in Algorithm 1.

To specify the algorithm, we need the following concept. Given an assignment x, let $J' \subseteq J$ be a set of clients such that $J' \supseteq \mathrm{clients}(x)$. We say that x is an α-*cover* w.r.t. J' if the following condition holds: if $\mathrm{load}_x(i) < \alpha \cdot c(i)$ for a base station i, then $N(i) \cap J' \subseteq \mathrm{clients}(x)$. In other words, a client from J' may not be assigned by an α-cover only if the load of each of its neighbors is at least an α fraction of its capacity.

The key step in Algorithm 1 below (Step 10) is to extend the assignment returned by the recursive call of Step 8. The algorithm maintains the invariant that the returned assignment is an α-cover w.r.t. J. Whenever the recursive call of Step 8 returns, the assignment is extended using the clients in J'' to ensure that the invariant holds true.

Algorithm 1 — CCC(I, J, c, d, p)

1: **if** $J = \emptyset$ **then** return empty assignment.
2: $J' = \{j \in J \mid p(j) = 0\}$
3: **if** $J' \neq \emptyset$ **then**
4: return CCC$(I, J \setminus J', c, d, p)$
5: **else**
6: $\delta = \min_{j \in J} \left\{ \frac{p(j)}{d(j)} \right\}$
7: for all j, define $p_1(j) = \delta \cdot d(j)$
8: $x \leftarrow$ CCC$(I, J, c, d, p - p_1)$
9: $J'' = \{j \in J \mid p(j) = p_1(j)\}$
10: using clients from J'', extend x to an α-cover w.r.t. J
11: return x
12: **end if**

The key to the analysis of the algorithm is the following result (see also [1, 8]).

Lemma 1. *Assume there exists some $\delta \in \mathbb{R}^+$ such that $p(j) = \delta \cdot d(j)$ for every client j. Consider any assignment x. If x is an α-cover w.r.t. J, then $p(\mathrm{clients}(x)) \geq (\frac{\alpha}{1+\alpha}) \cdot p(\mathrm{clients}(y))$ for any feasible assignment y.*

Proof. Let $S = \mathrm{clients}(x)$, and let $Y = \mathrm{clients}(y)$. Then

$$\begin{aligned} p(Y) &= p(Y \cap S) + p(Y \cap \overline{S}) \\ &= \delta \left[d(Y \cap S) + d(Y \cap \overline{S}) \right] \\ &\leq \delta \left[d(S) + c(N(\overline{S})) \right] \\ &\leq \delta \left[d(S) + d(S)/\alpha \right] \\ &= \frac{\alpha + 1}{\alpha} \cdot p(S) \ , \end{aligned}$$

where the first inequality follows from the feasibility of y and the definition of $N(\overline{S})$ (see Figure 1), and the second inequality follows from our assumption that x is an α-cover w.r.t. J. □

We note that the above lemma actually bounds the profit of an α-cover even with respect to *fractional* assignments, where a client may be covered by several base stations (so long as the profit is obtained only from fully covered clients).

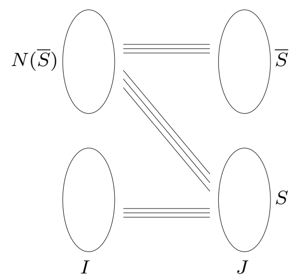


Fig. 1: Depiction of a solution S that uses an α fraction of the capacity of $N(\overline{S})$.

The following theorem shows that Algorithm cCC produces an $\frac{\alpha}{\alpha+1}$-approximation, assuming one can extend a given solution to an α-cover.[3]

Theorem 1. *Algorithm* cCC *returns an* $\frac{\alpha}{\alpha+1}$*-approximation.*

Proof. The proof is by induction on the number of recursive calls. The base case is trivial. For the inductive step, we need to consider two cases. For the cover returned in Step 4, by the induction hypothesis it is an $\frac{\alpha}{\alpha+1}$-approximation w.r.t. $J \setminus J'$, and since all clients in J' have zero profit, it is also an $\frac{\alpha}{\alpha+1}$-approximation w.r.t. J. For the cover returned in Step 11, note that by the induction hypothesis, the solution returned by the recursive call in Step 8 is an $\frac{\alpha}{\alpha+1}$-approximation w.r.t. profit function $p-p_1$. Since every client $j \in J''$ satisfies $p(j) - p_1(j) = 0$, it follows that any extension of this solution using clients from J'' is also an $\frac{\alpha}{\alpha+1}$-approximation w.r.t. to $p-p_1$. Since the algorithm extends this solution to an α-cover by adding clients from J'', and p_1 is proportional to the demand, by Lemma 1 we have that the extended α-cover is an $\frac{\alpha}{\alpha+1}$-approximation w.r.t. p_1. By the Local-Ratio Lemma (see, e.g., [14]), it follows that this solution is an $\frac{\alpha}{\alpha+1}$-approximation w.r.t. p, thus completing the proof. □

A closer examination of the local-ratio framework presented above shows that what the algorithm essentially does is to traverse the clients in non-decreasing order of their profit-to-demand ratio, while ensuring that any point, the current solution is an α-cover w.r.t. clients considered so far.

In what follows, we build upon the above framework, and show that given any $\gamma \in (0, 1]$, one can emulate distributively the above approach, while losing a mere $(1-\gamma)$ factor in the approximation guarantee. Furthermore, we show that this can be obtained in time $O(\gamma^{-2} \log^3 n)$.

3 A Distributed Approach

In this section we present a distributed algorithm for the CC problem. We first give an overview of our approach, and then turn to provide the details of our algorithm.

[3] In [1] they extend the solution to a maximal solution (w.r.t. set inclusion), which implies a $(1-r)$-cover.

3.1 Overview

Conceptually, the algorithm is derived by a series of transformations that allows us to represent any instance of CC as a multiple instances of maximal matching, which can be solved efficiently in a distributed model. Specifically, our generalization proceeds as follows.

Unit demand, unit capacity, unit profit. First, consider the case where all demands, capacities and profits are one unit each. In this case, CC is exactly equivalent to the problem of maximum matching, which for any $\varepsilon \in (0, 1])$ can be solved in $O(\frac{\log n}{\varepsilon})$ rounds with approximation ratio $(1 - \varepsilon)$ [12].

Unit demand, different capacities, unit profit. Next, suppose that all demands and profits are equal, but capacities may vary. This case is easy to solve as follows: Each base station i of capacity c can be viewed as c unit-capacity "virtual" base stations $i_1, \ldots, i_c$; then maximum matching can be applied to the graph consisting of the original clients and virtual base stations, where each client j originally connected to a base station i is now connected to all the induced virtual base stations $i_1, \ldots, i_c$. Some care needs to be exercised to show that this emulation can be carried out using $O(\log n)$ bit messages without any increase in the running time.

Different demands, different capacities, profit equals demand. The next generalization is to consider the case where base stations have arbitrary capacities, and clients have arbitrary demands, but the profit from each client is proportional to its demand. To solve this case, we use a scaling-like method: we round each demand to the nearest power (from above) of $1 + \varepsilon$, where $\varepsilon > 0$ is a given parameter. This rounding reduces the number of different demands, and as we show, only the $O(\log_{(1+\varepsilon)} n) = O(\frac{\log n}{\varepsilon})$ topmost different demands need be considered. For each fixed demand, we are back in the previous case. The penalty of rounding the profits is a $(1 + \varepsilon)$ factor degradation in the approximation ratio. Some additional complications due to interference between the different demands degrade the approximation ratio to $\frac{1-r}{2-r}$, where r is the maximal demand-to-capacity ratio in the given instance. As mentioned above, the running time increases by a factor of $O(\frac{\log n}{\varepsilon})$.

Different demands, different capacities, different profit. This last generalization is taken care of by applying the local ratio method presented in Section 2, which means that we need to go over the different profit-to-demand ratios in decreasing order. To avoid too many such ratios, we again use the trick of rounding to the nearest power of $(1 + \varepsilon)$, but this time we round the profits, resulting in an additional degradation of $(1 + \varepsilon)$ factor in the approximation ratio, and an additional factor of $O(\frac{\log n}{\varepsilon})$ in the running time.

3.2 Partitioning the Clients

Recall that all demands and capacities are integral, and assume that the minimum demand is 1. Let $\varepsilon \in (0, 1]$.

First Cut: Partition by Cost-Effectiveness. We consider a partition of the clients into sets $J_0, J_1, \ldots$, such that a client j is in J_k iff $\frac{p(j)}{d(j)} \in [(1 + \varepsilon)^k, (1 + \varepsilon)^{k+1})$. If these ratios are polynomially bounded, then we have $O(\log_{(1+\varepsilon)} n) = O(\frac{1}{\varepsilon} \log n)$ such sets. Denote the number of these sets by W. For every client j, we let its $(1+\varepsilon)$-*rounded profit* be defined by

$$p_{(1+\varepsilon)}(j) = \min_{k \in \mathbb{N}} \left\{ (1+\varepsilon)^k d(j) \mid (1+\varepsilon)^k d(j) \geq p(j) \right\} .$$

The following lemma relates the value of any solution to an instance of the r-CC problem, to the value of the same solution when considering the instance with $(1+\varepsilon)$-rounded profits.

Lemma 2. *Given some input $\mathcal{I} = (I, J, E, c, d, p)$ to the r-CC problem, and some $\varepsilon > 0$, consider the instance $\mathcal{I}' = (I, J, E, c, d, p_{(1+\varepsilon)})$, and let x be any assignment of clients. It follows that* $\text{clients}(x)$ *is a feasible solution to $\mathcal{I}$ iff it is a feasible solution to $\mathcal{I}'$, and* $p_{(1+\varepsilon)}(\text{clients}(x)) \leq (1+\varepsilon)\, p(\text{clients}(x))$.

Proof. The first part of the claim follows from the fact that feasibility is not affected by the change in the profit function, since it relies solely on the underlying topology of G, along with the base stations' capacities and clients' demands. For the second part, by the definition of $p_{(1+\varepsilon)}$ it follows that for every client j, $p_{(1+\varepsilon)}(j) \leq (1+\varepsilon)\, p(j)$, and therefore by considering sets of clients, the claim follows. $\square$

The following corollary is an immediate consequence of Lemma 2.

Corollary 1. *For every $\beta \leq 1$, and any instance $\mathcal{I} = (I, J, E, c, d, p)$, if a feasible assignment x is a β-approximate solution with respect to profit function $p_{(1+\varepsilon)}$, then it is a $\frac{\beta}{(1+\varepsilon)}$-approximate solution with respect to profit function p.*

Proof. Consider the instance $\mathcal{I}' = (G, c, d, p_{(1+\varepsilon)})$. First note that by Lemma 2, S is a feasible solution to $\mathcal{I}$. Furthermore, for any optimal solution S^* to $\mathcal{I}$, S^* is also a feasible solution to $\mathcal{I}'$, and since for any j we have $p_{(1+\varepsilon)}(j) \geq p(j)$ it follows that

$$\text{OPT}(\mathcal{I}) \leq \text{OPT}(\mathcal{I}') \leq \frac{1}{\beta} \cdot p_{(1+\varepsilon)}(S) \leq \frac{1+\varepsilon}{\beta} \cdot p(S) \ ,$$

as required, where the last inequality follows from lemma 2. $\square$

Corollary 1 ensures that by assuming that in every set J_k, the profits are of the same proportion as the demands, does not cause us to lose more than a $\frac{1}{(1+\varepsilon)}$ factor of the original profit obtained from the same solution.

We henceforth assume that the actual profit of every client j is its $(1+\varepsilon)$-rounded profit. It follows that in every J_k all clients have profits which are proportional to the demand. Note that in such a case, the order implied on the set of clients by their profit-to-demand ratio is exactly the same as the order in which CCC considers the clients, also assuming $(1+\varepsilon)$-rounded profits.

Second Cut: Partition by Demand. For every k, we consider a subpartition of the set J_k into subsets $J_k^0, J_k^1, \ldots$ such that a client $j \in J_k$ is in J_k^ℓ if $d(j) \in [(1+\varepsilon)^\ell, (1+\varepsilon)^{\ell+1})$. For every k we let r_k denote the maximal ℓ such that $J_k^\ell \neq \emptyset$. We further let $J'_k = \bigcup_{\ell \geq r_k - 3\log_{(1+\varepsilon)} n} J_k^\ell$.

3.3 A Distributed Algorithm

We now turn to describe our distributed algorithm, DCC. Let $\alpha = \frac{1-r}{1+\varepsilon}$. The goal of our algorithm is to produce an assignment that is an α-cover with respect to $J' = \bigcup_k J'_k$. Using the results presented in Section 2 this would serve as a first component in proving our approximation guarantee. We later show that by restricting our attention to J' we lose a marginal factor in the approximation ratio.

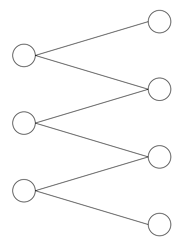
The algorithm, whose formal description in given in Algorithm 2, works as follows. It traverses the subsets J_k in decreasing order of k. For each k, it computes an α-cover with respect to J'_k (this is done by using Algorithm MC which we discuss in the following section). This enables us to show that DCC also produces an α-cover with respect to J'. For clarity we first analyze the performance of Algorithm DCC assuming that MC indeed produces an α cover with respect to J'_k, and then discuss the details of Algorithm MC. The following lemma shows that this assumption on MC suffices in order for DCC to produce an α-cover with respect to J'.

Algorithm 2 — DCC(I, J, c, d, p)

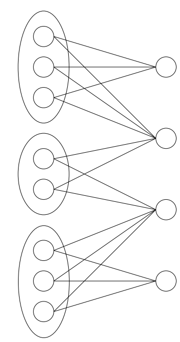
$R = J$ ▷ uncovered eligible clients
$x \leftarrow$ empty assignment
for every $i \in I$, let $\mathrm{res}_x(i) = c(i)$ ▷ the residual capacity
for $k = W$ downto 0 **do**
    $(x_k, \mathrm{res}_x) \leftarrow \mathrm{MC}(k, I, J_k \cap R, \mathrm{res}_x, d)$ ▷ compute an α-cover w.r.t. J'_k
    update x according to x_k
    remove all clients matched in x_k from R
    remove all ineligible clients from R
end for
return x

Lemma 3. *Assume that for every k,* MC *produces an α-cover with respect to J'_k. For every k, the cover produced by* DCC *after the end of the kth iteration, is an α-cover with respect to $J'_{\geq k} = \bigcup_{t \geq k} J'_k$.*

Proof. The claim follows from the fact that in every iteration, the residual capacity never increases, which implies that for any k, if we extend an α-cover for $J'_{\geq k}$, and ensure the extension is an α-cover for J'_{k-1}, then we obtain an α-cover for $J'_{\geq k-1}$. □



(a) Original instance.



(b) Virtual base-stations instance.

Fig. 2: The original graph representing $\mathcal{I}$ and its corresponding virtual base-stations graph representing $\mathrm{VG}(\mathcal{I})$. The clients are placed on the right.

3.4 Covering Equally Cost-effective Clients

In order to describe our algorithm, we need the following notion. Consider any $\varepsilon \in (0,1)$, and an instance $\mathcal{I} = (I, J, E, c, d, p)$ to the CC problem. Assume there exist some μ such that for all clients j, $d(j) \in [\frac{\mu}{1+\varepsilon}, \mu]$. We consider the *virtual base-stations instance* $\mathrm{VG}(\mathcal{I}) = (I', J, E', c', d, p)$, where every base station $i \in I$ is replaced by $\lfloor c(i)/\mu \rfloor$ base stations in I', each with capacity μ. We refer to these new base stations as *copies* of i. E' contains all virtual edges implied by the above swap, i.e., for every $(i,j) \in E$, we have an edge (i', j) for every copy i' of i. Given such an instance $\mathrm{VG}(\mathcal{I})$, note that every copy has sufficient capacity to cover any single client, and at most one such client. We may therefore assume without loss of generality that all demands are unit demands, and all capacities (of the copies) are unit capacities. It follows that any matching in $\mathrm{VG}(\mathcal{I})$ induces a feasible assignment of clients to base stations. See Figure 2 for an outline of a virtual instance corresponding to an original instance.

Given J_k, the goal of algorithm MC, whose formal description appears in Algorithm 3, is to produce an assignment that is an α-cover with respect to J'_k.

In what follows, we refer to MM as any distributed algorithm for finding a maximal matching in an unweighted bipartite graph. As mentioned earlier, the currently best algorithm for this problem is due to [3], which finds a maximal matching (with high probability) in expected logarithmic time.

Intuitively, the algorithm works as follows. For every base station i, the base station traverses $3\log_{(1+\varepsilon)} n$ subsets J_k^ℓ in decreasing order of ℓ, starting from the maximal ℓ for which it has eligible neighbors in. For every such J_k^ℓ, the algorithm considers its corresponding virtual base-stations instance G_k^ℓ while taking into account only eligible clients. It then computes distributively a maximal matching in the above graph using algorithm MM. Any matched client is assigned to its matched base station, and each base station updates its residual capacity accordingly.

Algorithm 3 — $\mathrm{MC}(k, I, J_k, c, d)$

1: $x_k \leftarrow$ empty assignment
2: for every $i \in I$, $\mathrm{res}_{x_k}(i) = c(i)$ ▷ the residual capacity
3: for every $i \in I$, let $r_k^i = \max\left\{\ell \mid \exists \text{ eligible } j \in N(i) \cap J_k^\ell\right\}$ ▷ every base station picks its highest relevant level
4: every base station i does:
5: **for** $\ell = r_k^i$ downto $r_k^i - 3\log_{(1+\varepsilon)} n$ **do** ▷ we only consider the topmost $3\log_{(1+\varepsilon)} n$ subsets
6: $\quad$ i announces to its eligible neighbors in J_k^ℓ about the round
7: $\quad$ $d_\ell^i \leftarrow$ maximal demand of an eligible client in $J_k^\ell \cap N(i)$
8: $\quad$ i uses $\left\lfloor \mathrm{res}_{x_k}(i)/d_\ell^i \right\rfloor$ copies of itself in the virtual graph G_k^ℓ
9: $\quad$ update x_k according to $\mathrm{MM}(G_k^\ell)$ ▷ performed in parallel
10: $\quad$ update $\mathrm{res}_{x_k}(i)$ ▷ update the residual capacity according to the demand of matched clients
11: **end for**
12: **return** $(x_k, \mathrm{res}_{x_k})$

We first show that the algorithm computes a $\frac{(1-r)}{1+\varepsilon}$-cover with respect to J_k'. In the sequel we show that by considering only the topmost $3\log n$ subsets we are able to obtain polylogarithmic running time in exchange for a marginal drop in the approximation ratio.

Lemma 4. *Algorithm* MC *computes a feasible* $\frac{1-r}{1+\varepsilon}$*-cover with respect to* J_k'.

Proof. We first note that the algorithm produces a feasible cover with respect to J_k'. To see this, note that any client j is assigned to a base station i only via the matching produced by MM. Since in the virtual base stations graph used by MM, we have $\left\lfloor \mathrm{res}_{x_k}(i)/d_\ell^i \right\rfloor$ copies of base station i, each with capacity d_ℓ^i, and every one of its neighbors in this round has demand at most d_ℓ^i (by Step 7), we are guaranteed that every base station has sufficient capacity to cover its matched clients in every round (since residual capacities are updated at the end of every round). We now turn to show the solution is indeed a $\frac{1-r}{1+\varepsilon}$-cover with respect to J_k'.

Consider any uncovered client $j \in J_k'$, and let $\ell \geq r_k - 3\log_{(1+\varepsilon)} n$ be such that $j \in J_k^\ell$. All we need to show is that for every $i \in N(j)$, i has used at least a $\frac{1-r}{1+\varepsilon}$-fraction of its capacity. Let i be any base station in $N(j)$. Note first that by maximality of r_k we have $r_k^i \leq r_k$, which implies that $\ell \geq r_k^i - 3\log_{(1+\varepsilon)} n$.

If i did not participate in a round corresponding to ℓ, this can only be because at that time, $\mathrm{res}_{x_k}(i) < d_\ell^i$.[4] Since $d_\ell^i \leq r \cdot c(i)$, this implies that $\mathrm{res}_{x_k}(i) < r \cdot c(i)$. It follows that in this case we are done.

Assume i did participate in a round corresponding to ℓ, and consider the copies of i in the virtual base stations graph. By Step 8, there are $\left\lfloor \mathrm{res}_{x_k}(i)/d_\ell^i \right\rfloor$ copies of i in this

[4] This is also the case when $\ell > r_k^i$.

graph, and they were all matched by MM since otherwise, we could have matched j, whose demand is at most d^i_ℓ, contradicting maximality of the output of MM.

The unused capacity due to rounding down the number of copies of i in the virtual graph implies the base station left out less than $d^i_\ell \leq r \cdot c(i)$ of its capacity from being used in this round. It follows that it has dedicated (and partly used) at least a $(1-r)$-fraction of its capacity for covering clients in all rounds up to (and including) round ℓ. Every client j covered by i in any such round $\ell' \leq \ell$, is matched to a copy of the base station, which represents a capacity of $d^i_{\ell'} \in [d(j), (1+\varepsilon)d(j))$. It follows that we effectively use up at least a $\frac{1}{1+\varepsilon}$-fraction of the capacity dedicated for covering clients in all rounds up to (and including) round ℓ. Combining the above we obtain that base station i has used at least a $\frac{1-r}{1+\varepsilon}$ of its capacity, as required. $\square$

Note that algorithm MC performs $3\log_{(1+\varepsilon)} n$ rounds, in each of which it executes Algorithm MM.

3.5 Wrapping Up

In this section we show how to combine the results presented in the previous sections, to obtain the following:

Theorem 2. *For every $\gamma \in (\frac{1}{n^2}, 1]$, algorithm* DCC *produces a $\frac{1-r}{2-r}(1-\gamma)$-approximate solution to r-CC, with high probability, in time $O(\gamma^{-2}\log^3 n)$.*

First we note that by combining Lemma 4 and Lemma 3, the solution produced by DCC is a $\frac{1-r}{1+\varepsilon}$-cover w.r.t. $J' = \bigcup_k J'_k$.

Let p_M be the maximal profit of any client in J. The following lemma shows that every $j \in J \setminus J'$, has very small profit.

Lemma 5. *Every $j \in J_k \setminus J'_k$ satisfies $p(j) \leq \frac{p_M}{n^3}$.*

Proof. Let $\ell < r_k - 3\log_{(1+\varepsilon)} n$ be such that $j \in J^\ell_k$. It follows that the demand of j is at most $(1+\varepsilon)^{r_k - 3\log_{(1+\varepsilon)} n} = \frac{(1+\varepsilon)^{r_k}}{n^3}$. It follows that $p(j) \leq \frac{p(j')}{n^3}$, where $j' \in J^{r_k}_k$. Note that by the definition of r_k, such a client j' exists. It follows that $p(j) \leq \frac{p_M}{n^3}$, as required. $\square$

We henceforth refer to a client j for which $p(j) \geq \frac{p_M}{n^3}$ as a *fat* client. Lemma 5 ensures that all fat clients are in J'. The following lemma shows that by ignoring non-fat clients, we lose only a negligible fraction of the possible profit.

Lemma 6. *Let* OPT *denote a solution to some instance $\mathcal{I}$ of r-CC, and let* OPT' *denote a solution to the same instance, restricted solely to fat clients. Then $p(\mathrm{OPT}) \leq (1+\frac{1}{n^2})p(\mathrm{OPT}')$.*

Proof. Let p_M denote the maximal profit of any client in J. Since clearly $p(\mathrm{OPT}) \geq p(\mathrm{OPT}') \geq p_M$, it follows that

$$\begin{aligned} p(\mathrm{OPT}) &= p(\mathrm{OPT} \cap J_B) + p(\mathrm{OPT} \cap \overline{J_B}) \\ &\leq p(\mathrm{OPT}') + n \cdot \frac{p_M}{n^3} \\ &= p(\mathrm{OPT}') + \frac{p_M}{n^2} \\ &\leq p(\mathrm{OPT}')(1 + \frac{1}{n^2}), \end{aligned}$$

as required. $\square$

The above lemmas ensure that any α-cover w.r.t. J' guarantees an approximation factor of $\frac{\alpha}{1+\alpha}(\frac{1}{1+1/n^2}) \approx \frac{\alpha}{1+\alpha} \cdot (1 - \frac{1}{n^2})$. Since we assume clients have $(1+\varepsilon)$-rounded profits, by Corollary 1 we lose at most an additional $\frac{1}{1+\varepsilon}$ factor in the approximation factor.

Given any $\gamma \in (\frac{1}{n^2}, 1]$, by considering an appropriate constant $\varepsilon = \varepsilon(\gamma) \in (0, 1]$, we can guarantee that the $\frac{1-r}{1+\varepsilon}$-cover produced by DCC is a $\frac{1-r}{2-r}(1-\gamma)$-approximate solution, thus completing the proof of Theorem 2.

As for the running time of algorithm DCC, we use the randomized algorithm of [3]. (This is the only place where randomization is used in our algorithm.) Running that algorithm for $c \log n$ rounds results in a maximal matching with probability at least $(1 - \frac{1}{n^{\Omega(c)}})$. Since the total number of times our algorithm invokes MM is at most n, by the Union Bound it follows that by choosing a sufficiently large constant c, we can guarantee, with high probability, that all executions of MM in Algorithm MC produce a maximal matching.

It follows that if $\max_j \{p(j)/d(j)\}$ is polynomially bounded, then our algorithm runs for $O(\gamma^{-2} \log^3 n)$ rounds, and produces a $\frac{1-r}{2-r}(1-\gamma)$-approximate solution with high probability.

A note on very small and very large values of γ. Note that for $\gamma < 1/n$, we can send the entire network information to every base station in time $O(\gamma^{-2}) = O(n^2)$, in which case every base station may calculate in a centralized manner a deterministic approximate solution. If we are given $\gamma > 1$, we use the algorithm with $\gamma = 1$.

4 Extensions

In this section we present several extensions of our results, whose proofs will appear in the full version.

Clients with Different Classes of Service and Location-Dependent Demands. Our algorithm can be extended to the model where every client j has a specific class of service q_j, and the profit from satisfying a demand d is $q_j d$. Moreover, the client may have a different demand from each possible base station (this may be the case when the requested service is location-dependent): namely, the demand client j has from base station i is $d(j,i)$, and the profit obtained from assigning client j to base station i is $p(j,i) = q_j \cdot d(j,i)$. We can show that even in this general setting, our algorithm provides the same approximation guarantees (with high probability), as well as having the same running time. We note that we still insist that on the condition that a profit is obtained from a client only if its demand is met in full (by one of its neighboring base stations).

Absence of Global Bounds. Our results extend to the model where base stations have no a priori knowledge of the maximal density of the instance (corresponding to the value W in our analysis), and perform on a merely local information basis. In this extension every base station takes on a myopic view of the network, and considers the partition corresponding solely to its neighboring clients. We can show that even in such an asynchronous environment, a simple extension of our algorithm provides the same approximation factors and time complexity.

5 Conclusions and Open Questions

In this work we presented a randomized distributed algorithm for the cellular coverage problem, such that for every $\gamma \in (0,1]$ our algorithm guarantees to produce a $\frac{1-r}{2-r}(1-\gamma)$-approximate solution with high probability, in time $O(\gamma^{-2}\log^3 n)$.

There are several interesting questions that arise from our work. First, our work provides a distributed emulation of a centralized local ratio algorithm. It is of great interest to see if similar emulations can be obtained to other problems, where local ratio algorithms provide good approximations. The main elements that seem to facilitate such an emulation are an ordering (or partitioning) of the input implied by the profit decomposition, and a notion of maximality that can be maintained locally. When considering the CC problem, a major goal is to try and improve the running time of a distributed algorithm for solving r-CC. There is currently no reason to believe that a good cover cannot be obtained in logarithmic time. Furthermore, it is interesting to see if there exists a distributed algorithm which makes use of the cover-by-many paradigm, which was shown to provide better solutions than the cover-by-one paradigm. For the more general formulation of the CC problem, it is not evident that one cannot obtain an approximation guarantee that is independent of r. In this respect, we conjecture that even for $r = 1$ the problem admits to constant approximation.

References

1. Amzallag, D., Bar-Yehuda, R., Raz, D., Scalosub, G.: Cell selection in 4G cellular networks. In: Proceedings of IEEE INFOCOM 2008, The 27th Annual Joint Conference of the IEEE Computer and Communications Societies. (2008) 700–708
2. Peleg, D.: Distributed computing: a locality-sensitive approach. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2000)
3. Israeli, A., Itai, A.: A fast and simple randomized parallel algorithm for maximal matching. Information Processing Letters **22**(2) (1986) 77–80

4. Hanly, S.V.: An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity. IEEE Journal on Selected Areas in Communications **13**(7) (1995) 1332–1340
5. Mathar, R., Schmeink, M.: Integrated optimal cell site selection and frequency allocation for cellular radio networks. Telecommunication Systems **21** (2002) 339–347
6. Sang, A., Wang, X., Madihian, M., Gitlin, R.D.: Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems. In: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MOBICOM). (2004) 302–314
7. Amzallag, D., Naor, J., Raz, D.: Coping with interference: From maximum coverage to planning cellular networks. In: Proceedings of the 4th Workshop on Approximation and Online Algorithms (WAOA). (2006) 29–42
8. Dawande, M., Kalagnanam, J., Keskinocak, P., Salman, F.S., Ravi, R.: Approximation algorithms for the multiple knapsack problem with assignment restrictions. Journal of Combinatorial Optimization **4**(2) (2000) 171–186
9. Shmoys, D.B., Tardos, É.: An approximation algorithm for the generalized assignment problem. Mathematical Programming **62** (1993) 461–474
10. Chekuri, C., Khanna, S.: A PTAS for the multiple knapsack problem. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). (2000) 213–222
11. Cohen, R., Katzir, L., Raz, D.: An efficient approximation for the generalized assignment problem. Information Processing Letters **100**(4) (2006) 162–166
12. Lotker, Z., Patt-Shamir, B., Pettie, S.: Improved distributed approximate matching. In: Proceedings of the 20th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA). (2008) 129–136
13. Bar-Yehuda, R., Even, S.: A local-ratio theorem for approximating the weighted vertex cover problem. Annals of Discrete Mathematics **25** (1985) 27–46
14. Bar-Noy, A., Bar-Yehuda, R., Freund, A., Naor, J., Shieber, B.: A unified approach to approximating resource allocation and schedualing. Journal of the ACM **48**(5) (2001) 1069–1090