Approximation Algorithms
for
Cellular Network Optimization

Christian Glaßer, Steffen Reith, and Heribert Vollmer

Theoretische Informatik
Universität Würzburg
Am Exerzierplatz 3
D-97072 Würzburg, Germany

{glasser, streit, vollmer}@informatik.uni-wuerzburg.de

Abstract

We consider two optimization problems for cellular telephone networks with interferences, which address the positioning of base stations (on given locations) with the aim to maximize the number of supplied demand nodes and minimize the number of stations that have to be built. It is known that in general, these problems are NP-complete and even hard to approximate. Here we show that the Euclidean versions of all problems allow a polynomial time approximation scheme (PTAS).

Furthermore, we consider some optimization problems without interferences in the Euclidean plane, addressing the problems to find first a minimum number of base stations locations and second to select a minimum set of given base station locations.

1 Introduction

In this paper we consider the following optimization problems for cellular telephone networks with interferences:

Maximize number of totally supplied nodes **(MTSN):**
Given is a set N of demand nodes (DN), a set B of potential locations for base stations (BS), a budget $k \in \mathbb{N}$ and an interference factor $\gamma \in \mathbb{Q}$ with $\gamma > 1$. Furthermore, for every $i \in B$ and $j \in N$, the variables $u_{i,j} \in \mathbb{Q}_+$ and $t_{i,j} \in \{0, 1\}$ are given. The variable $t_{i,j}$ expresses if a demand node is supplied by a base

station: DN j is supplied by BS i if and only if $t_{i,j} = 1$. The variable $u_{i,j}$ gives us the intensity of the signal of BS i in DN j (if we would build this station).

If DN j is supplied by BS i, we say it is totally supplied by BS i if and only if the corresponding signal in DN j is essentially stronger than the sum of all other (interfering) signals in this demand node, formally:

$$u_{i,j} > \gamma \cdot \sum_{\substack{\text{built BS } k \\ k \neq i}} u_{k,j}$$

Now the aim is to find a collection of at most k base stations, such that the number of totally supplied demand nodes is maximal.

Minimize number of base stations **(MBS):**
Here we have almost the same situation as in the problem MTSN, with the difference that a number ℓ is given instead of the budget k. The task is to find a minimum number of base stations such that at least ℓ demand nodes are totally supplied.

In both problems we consider interferences with signals from other base stations as they appear, e.g., in TDMA (and to a lesser extend in CDMA) based networks.

It is known that the above problems are (as decision problems) NP-complete and their optimization versions are even hard to approximate [Gla98]. In this paper we want to consider the following special cases:

By EMTSN (EMBS) let us denote the subproblem of MTSN (MBS, resp.) where we make the following assumptions:

- the demand nodes and the base stations are located in the Euclidean plane;
- the activity of the base stations is local (i.e. the intensity of the signal is 0 outside a range of action D);
- there exists a minimal distance d between all base stations that have to be built.

A result of this paper is that these geometric problems have polynomial time approximation schemes, i.e., for any given performance ratio ε, there is a polynomial time approximation algorithm.

In possible future technologies where interferences play no or not such an important role as today, our algorithms above will of course work as well. Therefore we define and examine two problems where by definition we neglect interferences, see Section 4. These problems may also be useful for an approximate planning of a present cellular network, at first disregarding interferences.

2 A Polynomial Time Approximation Scheme for EMTSN

The following algorithm constitutes an approximation scheme for EMTSN in polynomial time.

Given are the sets N of demand nodes, B of base station locations, budget k, interference factor γ and variables $t_{i,j}$, $u_{i,j}$ for every $i \in B$, $j \in N$. Since we consider the Euclidean version of MTSN, we have additional information on the location of demand nodes and base stations in the Euclidean plane, the range of action D and the minimal distance d between base stations.

We fix a number l, the so called shifting parameter. This number will later determine the performance ratio of our algorithm. By increasing l we will get better approximations.

1. Let the set N of the given demand nodes in the plane be enclosed in the rectangular area A.

2. Fix a division of A into horizontal and vertical strips of width D, where each strip is left (up) closed and right (down) open. By considering groups of l consecutive strips, we obtain a grid of width lD. It is clear that there are l^2 different such ways of partitioning A into strips of width lD. We denote the shift partitions that result, by $S_1, \ldots, S_{l^2}$. Notice that such a partition can be described by the coordinates of a single point in the plane. Therefore, we can fix a partitioning of A in polynomial time even if area A is large.

3. Each partition S_i consists of several squares of size $(lD)^2$. Notice that the number of such squares is possibly superpolynomial in the input size, but in this case almost all squares do not contain any demand node. Therefore, it suffices to consider the small number (polynomially bounded in the input size) of non-empty squares, i.e., squares containing at least one demand node. This can be done efficiently because it is possible to keep all demand nodes in a suitable data structure, such that the corresponding square can be easily determined by the position of the demand node in the Euclidean plane. In each such square s we delete all demand nodes that are located in the border-strips of width D; thus we obtain a new square s' of the same size but with no demand nodes located in distance less than D to the border. For $0 \leq j \leq |B|$ we want to solve the local maximization problem in s', i.e., we maximize the number of totally supplied demand nodes in s', where we are allowed to build up to j base stations in s'. We can do this by exhaustive search, because we can build at most $c =_{df} 4(lD)^2/(\pi d^2)$ base stations; this follows by the simple fact that circles of radius $d/2$ around every base station are not allowed to intersect because of the minimal distance d between two base stations. A more careful analysis shows that even $c =_{df} 2(lD)^2/(d^2\sqrt{3})$ is sufficient; however, the only important point is that c is a constant. Thus,

for each square s, we obtain a $(c+1)$-dimensional vector v_s, where the j-th component is the maximal number of totally supplied demand nodes in s if we are allowed to build at most j base stations.

4. Now we have reached another optimization problem, which is a kind of knapsack-problem. We are given a list L of $(c+1)$-dimensional vectors $v_s = (v_{s,0}, \dots, v_{s,c})$ of natural numbers (one vector for each non-empty square s) and a maximal weight k (our budget). We have to find a total mapping $f\colon \{1, \dots, |L|\} \to \{0, \dots, c\}$ giving us for each square, which contains at least one demand node, the number of base stations to be built such that $\sum_{1 \le i \le |L|} f(i) \le k$ and $\sum_{1 \le i \le |L|} v_{i,f(i)}$ is maximal. In other words, we want to distribute at most k base stations to $|L|$ squares, such that the number of totally supplied demand nodes is maximal. Since the $v_{i,j}$ are small numbers, we can solve this problem in polynomial time with the technique of dynamic programming as we describe next (this technique is a slight modification of an algorithm solving the knapsack problem which can be found in [Pap94] pp. 203).

   Let

   $$W(j,v) = \min\Big\{ \sum_{1 \le i \le j} f(i) \;\Big|\; f\colon \{1, \dots, j\} \to \{0, \dots, c\} \text{ and } \sum_{1 \le i \le j} v_{i,f(i)} = v \Big\}$$

   for $0 \le v \le |N|$ and $1 \le j \le |L|$.

   So the function $W(j,v)$ gives us the minimal number of base stations located in the first j squares such that exactly v demand nodes can be supplied totally.

   Let $W(0,v) = \infty$ for all $0 \le v \le |N|$. It is easy to see that the following recursive procedure determines these values: Let $0 \le v \le |N|$ and $0 \le j < |L|$; then we set

   $$W(j+1, v) = \min\{\, W(j, v - v_{j+1,k}) + k \;\mid\; 0 \le k \le c \,\}.$$

   Therefore, we can compute all values $W(k,v)$ in time $O(|N| \cdot |L|)$.

   Now we can search the largest $v_{\max}$ such that $W(|L|, v_{\max}) \le k$; this is the solution of the knapsack-problem.

5. For the base station problem this means that we can supply $v_{\max}$ demand nodes totally if we build at most k base stations. Hence each partition S_i induces a number of demand nodes being totally supplied by at most k base stations. Under all these solutions we choose one having the largest number of totally supplied demand nodes.

Now we want to show that the relative error of this algorithm is at most $4/l$. We fix an optimal solution of the given problem instance I. On the one hand, for each demand node there exist exactly $4l-4$ partitions in which this demand node is located on a border-strip. On the other hand there are l^2 partitions at all. By the pigeon hole principle we obtain that there exists a partition S' such that the following holds: At most a fraction of $(4l-4)/l^2 \leq 4/l$ of the totally supplied demand nodes (of the optimal solution) are located on a border-strip. If we delete all demand nodes on all border-strips of S' we obtain a new instance I' of our problem. It is easy to see that the solution for S' obtained by local optimization (first optimize each single square, then solve the knapsack-problem) equals the solution obtained by global optimization, because we have deleted all demand nodes located outside a square and inside the range of action of a base station in this square. Thus base stations in one square cannot influence demand nodes in another square. Therefore, we find in polynomial time an optimal solution for problem instance I'.

From the optimal solution for I we obtain a solution for I' if we hide the (small amount of) totally supplied demand nodes located on the border-strips of S'. This gives $\mathrm{OPT}_I \leq \mathrm{OPT}_{I'}/(1-4/l)$, which shows that our approximated solution is near to the optimal solution. Since every optimal solution for I' induces a solution for I, we have:

$$\mathrm{OPT}_{I'} \leq \mathrm{OPT}_I \leq \mathrm{OPT}_{I'}/(1-4/l)$$

Therefore, at least for partition S' our algorithm finds a solution having a relative error $\leq 4/l$, hence the overall solution of the algorithm will not have a larger error.

Since the l in the above argument is a number which we can choose beforehand, we see that for every performance ratio $\varepsilon > 0$ we obtain a polynomial algorithm for the EMTSN problem.

3 Approximating EMBS in Polynomial Time

Let us now turn to the EMBS problem. We take over the notations from the problem EMTSN, with the difference that ℓ, the minimal number of totally supplied demand nodes, is given (instead of budget k). Since EMBS is a minimization problem, where we demand a certain amount of success (at least ℓ totally supplied demand nodes), it is possible that there exist problem instances which do not have solutions. Even if an instance has a solution, it could happen that this is the only one. In this case, every approximation algorithm ought to find the optimal solution. So in our opinion we can not hope to find an approximation algorithm for this strong problem, so we have to weaken to demand for exactly ℓ demand nodes:

Starting from the PTAS for EMTSN, for every ε we construct an algorithm A_ε such that the following holds: For every instance I, which asks for ℓ totally supplied demand nodes and which has an optimal solution of b base station, our algorithm gives a solution with $b' \leq b$ base stations and $\ell' \geq \ell/(1+\varepsilon)$ totally supplied demand nodes.

Let I be an instance with a set of base stations B and a set of demand nodes N, such that b is the minimal number of base stations which have to be built if we want to supply at least ℓ demand nodes totally. Let B_ε be the approximation algorithm for EMTSN whose solutions have a relative error $\leq \varepsilon$. On input I, Algorithm A_ε applies B_ε on instance I with budget $k = 1, \ldots, |B|$ and obtains a list of numbers of totally supplied demand nodes $\ell_1, \ldots, \ell_{|B|}$. Let i be the smallest number with $1 \leq i \leq |B|$ such that $\ell_i \geq \ell/(1+\varepsilon)$. It follows $j < b$ for all $1 \leq j < i$; otherwise we could supply at least ℓ demand nodes totally by at most j base stations, thus $\ell_j = B_\varepsilon(I, j) \geq \ell/(1+\varepsilon)$, a contradiction. Therefore, on instance I with budget i, Algorithm B_ε outputs a collection of $b' \leq b$ base station which supply $\ell' \geq \ell/(1+\varepsilon)$ demand nodes totally. Hence Algorithm A_ε outputs the solution of B_ε on instance I with budget i.

4 Optimizing Networks without Interferences

4.1 A Polynomial Time Approximation Scheme for Base Station Localization

We define the following optimization problem in the Euclidean plane:

Minimize number of base station locations **(**EBSL**):**
Given is the set N of demand nodes located in the Euclidean plane. The task is to compute a minimal cardinality set of locations for base stations such that all demand nodes are supplied; where here supplied means that the DN is within distance D from a built base station.

In this problem we hence do not fix a set of potential locations for the base stations; the goal consists in computing these potential locations in an optimal way.

Let us now turn to an approximation algorithm for the EBSL problem. The algorithm we describe below is an immediate translation of the algorithm for the disk covering problem presented in [HM85] to case of the Euclidean plane. Let l be any number.

1. Let the demand nodes be enclosed in the plane in area A. As in the algorithm for EMTSN, construct shift partitions $S_1, \ldots, S_{l^2}$, where each partition divides the plane into a grid of squares of size $lD \times lD$.

2. For each partition S_i, locally solve the maximization problem for each non-empty square s in S_i. The solution for S_i then is obtained by the union of all base stations over all non-empty squares s.

3. Under all solutions for the different S_i, choose the minimum.

This algorithm works in polynomial time, and solves the EBSL problem with performance ratio at most $(1 + 1/l)^2$ (see [HM85]).

4.2 A PTAS for EMBS$_{NI}$

Finally, let us consider the problem EMBS, but this time neglecting possible interferences.

Minimize number of base stations without interferences **(EMBS$_{NI}$):** Given is a set N of demand nodes, a set B of potential locations of base stations, and a number ℓ. The task is to find a minimum number of base stations such that at least ℓ demand nodes are supplied, where here, supplied simply means that the demand nodes are within distance D from a built base station.

Note that in contrast to the previous subsection this time potential base station locations are given. Again, we present a polynomial time approximation scheme.

Given is a set N of demand nodes, a set B of potential locations of base stations, and a number ℓ.

1. Let the demand nodes be enclosed in the plane in area A. As in the algorithm for EMTSN, construct shift partitions $S_1, \ldots S_{l^2}$, where each partition divides the plane into a grid of squares of size $lD \times lD$.

2. Fix a partition S_i. We locally solve a number of maximization problems for each square s in S_i as follows: For each number j compute the maximal number $v_{s,j}$ of demand nodes in s that can be supplied, when we are allowed to build j base stations. Here, base stations can be build either within s or within the border strips of neighboring square. Let us denote by $U(s)$ the square s plus these border strips, hence $U(s)$ is a square of size $(l+2)\times(l+2)$. Hence, the task is to supply a maximal number of nodes within s by building j base stations in $U(s)$.

3. Now we have a list L of $(c+1)$-dimensional vectors $v_s = (v_{s,0}, \ldots, v_{s,c})$ of natuaral numbers (one vector for each square s; the c is as above in the algorithm for EMTSN). The entry $v_{s,j}$ gives us the number of nodes in s that can be supplied by building j base stations in $U(s)$. The problem we have to solve now is to choose for each square s a number $v_{s,j}$ such that the sum of all these is at least ℓ, while the number of base stations that have to be build (i.e., the number of j's over all picked $v_{s,j}$) is minimal. This is a

knapsack problem which we solve using a dynamic programming approach, analogous to the algorithm for EMTSN.

4. Finally, among the solutions for the partitions $S_1, \ldots, S_{l^2}$ we output the one with the minimal solution.

Let us analyze the error of this algorithm. Consider a problem instance I and an optimal solution for I. The local optimizations in the squares s of each partition will not lead to a larger number of base stations build in $U(s)$ than in the optimal solution, because we restrict ourselves to a subset of the demand nodes that have to be supplied. Consider now the sum of all these local solutions over all squares s. The sum of all base stations within s is by the above no more than OPT_I, however, we will possibly add up base stations in the border strips $U(s) - s$ that may not be necessary. There are l^2 partitions; over all partitions, each base station can be located at most $4l$ times in a border strip, hence, by the pigeon hole principle, there must be a partition where the ratio of stations in border strips is at most $4/l$. The solution found in this partition can therefore be not larger than $\mathrm{OPT}_I(1 + 4/l)$.

5 Conclusion

We presented polynomial time approximation schemes (PTAS) for the optimization problems EMTSN, EBSL and EMBS_{NI}. Furthmore we gave a similar family of algorithms for EMBS. This means that for any performance ratio ε we wish, there is a polynomial time approximation algorithm for all four problems. One might ask is if there is a uniform approximation scheme, working for all ratios ε with a runtime which is polynomial in the input length and $1/\varepsilon$. (This would mean that the problems have a so called fully polynomial time approximation scheme (FPTAS), see[CK99].) However, for the EBSL, EMBS_{NI} and EMBS problems such stronger schemes cannot exists (unless P = NP). For the EBSL problem, this is due to the fact that the corresponding decision problem is strongly NP-complete and optimization problems for such problems cannot have FPTASs [GJ79, Theorem 6.8] (this was observed already in [HM85]). Since EBSL reduces to EMBS_{NI} and EMBS, the same holds here. We suspect that the decision version of EMTSN is strongly NP-complete as well, hence also for this problem, a PTAS as we presented here is the best we can hope for.

It should be remarked that the algorithm we gave above for the EMTSN problem can in fact be generalized: Our algorithm maximized the number of *totally supplied nodes*. In fact, "totally supplied" here can be replaced by any other, even much stronger, condition — let us call it *super-supplied*; as long as it can be checked in polynomial time whether a node is super-supplied (by a given collection of base stations), our algorithm will produce an approximated solution in polynomial time. This allows us, e.g., to take peculiarities of the landscape (highways, skyscrapers, forests, etc.) into account. Note that we used the fact that the rectangular area A is embedded in the Euclidean plane *only* to determine the positions of base stations and demand nodes. None of the algorithms above make any assumption about a metric for the field strength/interferences.

As a final remark, we want to note that our algorithm is ideally suited for parallelization. First, the results for the l^2 partitions can be computed in parallel; second, even for a fixed partition the local optimization problems within each square s can also be performed in parallel.

Acknowledgment. We are grateful to Kenji Leibnitz, Kurt Tutschku, and Klaus W. Wagner for helpful discussions.

References

[CK99] P. Crescenzi and V. Kann. A compendium of NP optimization problems. URL: http://www.nada.kth.se/∼viggo/problemlist/compendium.html, 1999.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* Freeman, New York, 1979.

[Gla98] C. Glaßer. On the approximability of problems for cellular networks. In *Center of Network Optimization – Activity Report 1997/98.* CNO Würzburg Trier Aachen, 1998.

[HM85] D. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM*, 32:130–136, 1985.

[Pap94] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, Reading, MA, 1994.