A PTAS for Delay Minimization in Establishing Wireless Conference Calls

Leah Epstein[1,⋆] and Asaf Levin[2]

[1] School of Computer Science, The Interdisciplinary Center, Herzliya, Israel
lea@idc.ac.il.
[2] Department of Statistics, The Hebrew University, Jerusalem, Israel
levinas@mscc.huji.ac.il

Abstract. A prevailing feature of mobile telephony systems is that the location of a mobile user may be unknown. Therefore, when the system is to establish a call between users, it may need to search, or page, all the cells that it suspects the users may be located in, in order to find the cells where the users currently reside. The search consumes expensive wireless links which motivates search techniques that page as few cells as possible.

We consider cellular systems with n cells and m mobile users roaming among the cells. The location of the users is uncertain and is given by m probability distribution vectors. Whenever the system needs to find specific users, it conducts a search operation lasting at most d rounds. In each round the system may check an arbitrary subset of cells to see which users are located there. The problem of finding a single user is known to be polynomially solvable. Whereas the problem of finding any constant number of users (at least 2) in any fixed (constant) number of rounds (at least two rounds) is known to be NP-hard. In this paper we present a simple polynomial-time approximation scheme for this problem with a constant number of rounds and a constant number of users. This result improves an earlier $\frac{e}{e-1} \sim 1.581977$-approximation of Bar-Noy and Malewicz.

1 Introduction

ESTABLISHING WIRELESS CONFERENCE CALLS UNDER DELAY CONSTRAINTS PROBLEM (EWCC) is concerned with establishing a conference call involving $m+1$ users (from which one has a static position and the other m users have dynamic locations) in a cellular network. The main property of a cellular network is that the users are roaming. This places another step in the process of establishment of the conference call. I.e., the system needs to find out to which cell each user is connected at the moment. Using historical data the system has a certain probability vector for each user that describes the probability that the

⋆ Research supported by Israel Science Foundation (grant no. 250/01).

G. Persiano and R. Solis-Oba (Eds.): WAOA 2004, LNCS 3351, pp. 36–47, 2005.
© Springer-Verlag Berlin Heidelberg 2005

system will find the user in each cell. We assume that each user is connected to exactly one cell in the system and that the locations of the different users are independent random variables.

In order to find a set of users the system may page a subset of cells. Each cell in this subset returns a complete and accurate list of all the users that are connected to it. We assume that the search lasts a short period and during this period users do not move from one cell to another. The search strategy is to page a certain subset of cells looking for the users that participate in the conference call. After the system gets the answers from all the paged cells, it decides whether it needs to continue to the next round (i.e., the search did not find all the users) or it can stop the search (i.e., all the users have been already found). In order to ensure a reasonable quality of service there is an upper bound on the maximum number of rounds, denoted by d. We assume that the system must find all the users within d rounds. Therefore, if the system does not find all the participating users within the first $d-1$ rounds then in the last round it must page all the cells it did not page before. In this paper we follow Bar-Noy and Malewicz [1], and restrict our search strategy to *oblivious* algorithms, in which the subset of cells that is paged in round i does not depend on the actual users that the system found in round $1, 2 \dots, i-1$. However, if the search process is completed at round i, the algorithm may stop. There are other search strategies that are known as *adaptive search strategies* in which the subset of cells that is paged in round i depends on the users that have been found so far. As noted in [1] the two versions coincide for the special case of two rounds.

The goal of EWCC is to minimize the expected number of cells that the system pages throughout the search.

If $d = 1$ then EWCC is trivial since the system must page all cells in the first round, and the solution costs n. If $m = 1$ EWCC can be solved in polynomial time using a simple dynamic programming [2, 3]. Bar-Noy and Malewicz [1] showed that EWCC is NP-hard for any pair of fixed values of m, d such that $m, d \geq 2$. They also presented an approximation algorithm with performance guarantee of $\frac{e}{e-1} \sim 1.581977$ for arbitrary values of d, m, and for the special (NP-hard) case where $d = m = 2$ they showed that their approximation algorithm is a $\frac{4}{3}$-approximation. Bar-Noy and Malewicz raised the open problem of the existence of a polynomial-time approximation scheme for EWCC. In this paper we give the first positive answer for this question by presenting a polynomial-time approximation scheme (PTAS) for the case of a fixed number of rounds and a fixed number of users (m and d are arbitrary constant integer values).

We now present a formal definition of EWCC. Denote the cell set by $C = \{1, 2, \dots, n\}$, and the user set by $U = \{1, 2, \dots, m\}$. For $i \in U$ and $j \in C$, denote by p_i^j the probability that user i is located at cell j. We assume that $p_i^j > 0$, $\forall i, j$. Given a positive matrix $P = \{p_i^j\}_{i,j}$ and a bound d on the number of rounds, a feasible solution is a partition $C_1, C_2, \dots, C_d$ of C with the interpretation that in round k the system pages the cells in the subset C_k, unless it has already found all the users. A partition $C_1, C_2, \dots, C_d$ induces probabilities $(P_k)_{k=1}^d$ where P_k

denotes the probability that the search will last for at least k rounds. I.e., P_k is the probability that $C_1 \cup C_2 \cup \cdots \cup C_{k-1}$ does not contain U. Then, the cost of $C_1, C_2, \ldots, C_d$ is $\sum_{k=1}^{d} P_i|C_i|$. The goal of EWCC is to find a partition of C that minimizes its cost.

We now give a more detailed expression for the cost of a partition for the special case of two rounds: $C_2 = C \setminus C_1$, denote $p_i(C_1) = \sum_{j \in C_1} p_i^j$, which is the probability to find user i in the first round. Therefore $P_2 = 1 - \prod_{i \in U} p_i(C_1)$, and the cost associated with the partition $C_1, C \setminus C_1$ is exactly $|C_1| + (n - |C_1|) \cdot \left(1 - \prod_{i \in U} p_i(C_1)\right)$.

We start the paper with a PTAS for two rounds, and later extend it to an arbitrary (constant) number of rounds d.

2 A PTAS for Two Rounds

In this section we present the main result of this paper; a polynomial time approximation scheme for EWCC when $d = 2$. We fix an optimal solution OPT. Our scheme is composed of two guessing steps. In these guessing steps we guess certain information about the structure of OPT. Each guessing step can be emulated via an exhaustive enumeration of all the possibilities for this piece of information. So our algorithm runs all the possibilities, and among them chooses the best solution achieved. In the analysis it is sufficient to consider the solution obtained when we check the right guess.

Given OPT, denote by OPT_1 the number of cells that OPT pages in the first round, and by α_i the probability that OPT does not find user i in the first round. Therefore, the cost of OPT denoted by $COST(OPT)$ is $COST(OPT) = OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in U}(1 - \alpha_\ell)\right)$.

Recall that m is a constant, and let ε be a value such that $0 < \varepsilon < \frac{1}{(m+1)}$. If $n \leq m$, then EWCC can be solved in a constant time via exhaustive enumeration (since m is a constant), and therefore we assume that $n > m$. Denote the probability intervals $I_0 = (0, \frac{\varepsilon}{n^2}]$, and for $1 \leq i \leq \left\lceil \log_{1+\varepsilon}\left(\frac{n^2}{\varepsilon}\right)\right\rceil$,

$$I_i = \left(\frac{\varepsilon}{n^2}(1+\varepsilon)^{i-1}, \frac{\varepsilon}{n^2}(1+\varepsilon)^{i}\right].$$

Our first guessing step guesses for each $\ell \in U$, the index $i(\ell)$ such that $\alpha_\ell \in I_{i(\ell)}$. The following lemma is trivial:

Lemma 1. *The number of possibilities for the first guessing step is*

$$O\left(\left[\log_{1+\varepsilon}\left(\frac{n^2}{\varepsilon}\right) + 2\right]^m\right) .$$

Therefore, performing an exhaustive enumeration for this guessing step can be done in polynomial time. We continue to analyze the iteration of this step in which we guess the right values that correspond to OPT. For all $\ell \in U$, we denote *the guess of* α_ℓ by β_ℓ to be the upper bound of $I_{i(\ell)}$; $\beta_\ell = \frac{\varepsilon}{n^2}(1+\varepsilon)^{i(\ell)}$.

The next step is to scale up the probabilities as follows: for all i, j define $q_i^j = p_i^j/\beta_i$ to be the *scaled probability of* i *and* j. We consider the vector $Q^j = (q_i^j)_{i\in U}$ of the scaled probabilities that the users are in cell j. We remove all cells with scaled probability larger than 1. Such cells cannot be paged in the second round, and therefore must be paged in the first round. We further assign a type and weight for each Q^j according to the following way. Let q_i^j be a maximum entry in Q^j, then we assign a weight $w^j = q_i^j$ to Q^j, and we define $\tilde{Q}^j = \left(q_\ell^j/w^j\right)_{\ell\in U}$. Note that $Q^j = w^j \cdot \tilde{Q}^j$. We define a set of intervals $\mathcal{J}$ as follows: $J_0 = (0, \varepsilon]$, and for all $k \geq 1$, $J_k = (\varepsilon \cdot (1+\varepsilon)^{k-1}, \varepsilon \cdot (1+\varepsilon)^k]$, and $\mathcal{J} = \{J_0, J_1, \ldots\}$. For each entry ℓ in $\tilde{Q}^j$, we find the interval from $\mathcal{J}$ that contains q_ℓ^j/w^j. We assign the type of Q^j to be the following vector. For each ℓ, compute a value t_ℓ such that $\frac{q_\ell^j}{w^j} \in J_{t_\ell}$, then the type of Q^j is the vector $(t_1, t_2, \ldots, t_m)$.

Lemma 2. *The number of possible types is* $O\left(\left[\log_{1+\varepsilon}\left(\frac{1}{\varepsilon}\right) + 2\right]^m\right)$.

Proof. To see this note that for all ℓ, j, we have $\frac{q_\ell^j}{w^j} \leq 1$. Therefore, it is enough to use the first $\lceil\log_{1+\varepsilon}\left(\frac{1}{\varepsilon}\right)\rceil + 1$ intervals in $\mathcal{J}$. The bound on the number of possibilities of types that our instance contains follows.

Note that the bound on the number of types is a constant (for fixed values of ε, m). Our second guessing step is to guess OPT_1 (since the first round is never skipped, this is an integer between 1 and n) and guess the *number* of cells from each type that OPT pages in the second round (this also gives the number of cells from each type that OPT pages in the first round).

Lemma 3. *The number of possible guesses is bounded by* $O\left(n^{\left[\log_{1+\varepsilon}\left(\frac{1}{\varepsilon}\right)+2\right]^m}\right)$.

Proof. The number of cells from each type is an integer between 0 and $OPT_2 = n - OPT_1 \leq n-1$. This guess also implies a guess for OPT_2 and OPT_1.

Note that the number of possibilities for this guessing step is polynomial (for fixed values of ε, m).

Assume that for a type $T = (t_1, t_2, \ldots, t_m)$, OPT has $OPT_2(T)$ cells of type T that are paged in the second round. We sort the cells of type T according to their weight, and we assign the second round the $OPT_2(T)$ cells (among the cells of type T) that have the least weight. We apply this procedure for all the types T. We would like to ignore all invalid solutions. In order to be valid, the probability bounds β_ℓ must be satisfied, i.e. the sum of (non-scaled) probabilities must be in the interval $I_{i(\ell)}$. We slightly relax this requirement since a result of the scaling may shift the sum out of this interval. Instead, we disregard probabilities such that their scaled probability is in the interval $(0, \varepsilon]$, and we allow the sum of the other (non-scaled) probabilities to reside in the interval $[0, \beta_\ell(1+\varepsilon)]$. Equivalently, the sum of scaled and rounded vectors of probability of chosen cells (ignoring the small components as explained above) should be such that no component exceeds $1+\varepsilon$. For some guesses we obtain a candidate solution. Among all candidate solutions we output the one whose cost is minimized.

Lemma 4. *For a fixed number of users m, and for a constant $\varepsilon > 0$, the above scheme takes polynomial time.*

Proof. By Lemmas 1,2 and 3, the number of possibilities in the first step and in the second step is polynomial. The time to compute the resulting candidate solution for a single guess is clearly polynomial (i.e., finding a maximum value for each cell and finding its weight is polynomial, and the rest is simply sorting of the cells according to their weights), and the time to compute its cost is also polynomial. Therefore, the scheme has polynomial running time.

3 Analysis

We analyze the iteration of the first guessing step in which the guessed values of β_i $\forall i$ are the right guesses. We also assume that in the second guessing step we guess the right value of OPT_1 and the right number of cells of each type that OPT pages in the second round. We analyze the cost of the corresponding candidate solution.

Lemma 5. *The right set of guesses leads to a candidate solution.*

Proof. We have to show that for each user i, the sum of the probabilities (when we ignore cells whose scaled probability is at most ε) of finding user i in the second round is at most $\beta_i(1+\varepsilon)$. For a type $T = (t_1, t_2, \ldots, t_m)$ with $t_i \geq 0$, OPT selects $OPT_2(T)$ cells of type T with sum of weights that is at least the sum of weights of the cells of type T that the candidate solution selects (note that the weights are not changed in the process of partitioning the cells into types). By definition of $\mathcal{J}$, the probabilities of having user i in a pair of cells of type T with the same weight, are within a multiplicative factor of $1+\varepsilon$. Therefore, the contribution of type T cells to the probability that the candidate solution finds i during the second round, is at most $1+\varepsilon$ times the contribution of type T cells to the probability that OPT finds i only during the second round. Since the probability that OPT finds i only during the second round is $\alpha_i \leq \beta_i$, the claim follows.

Lemma 6. *Consider a user i, then the probability that the candidate solution finds i during the second round (and not during the first round) is at most $\beta_i(1+(m+1)\varepsilon)$.*

Proof. Consider a type $T = (t_1, t_2, \ldots, t_m)$. First, assume that $t_i \geq 1$. By Lemma 5, the contribution of type T cells to the probability that the candidate solution finds i during the second round, is at most $1+\varepsilon$ times the contribution of type T cells to the probability that OPT finds i only during the second round.

Next, consider a type T such that $t_i = 0$. For such a type we define the *leader* of T to be the first entry of the type vector that relates to the largest interval (the interval which contains the point 1). There exists at least one such entry as in $\tilde{Q}^j$ there is at least one unit entry. Note that the sum of scaled probabilities of finding user ℓ of all the cells paged by OPT with a type such that ℓ is the leader, is at most $1+\varepsilon$. Therefore, the total contribution of scaled probabilities of all the cells of any type T such that $t_i = 0$ and ℓ is acting as the leader of T is at most $(1+\varepsilon)\varepsilon$. Summing over all ℓ (note that $\ell \neq i$), we get an increase of $(m-1)\varepsilon(1+\varepsilon)$ caused by the types where $t_i = 0$. In terms of the original probabilities (i.e., for each cell we multiply its probability by β_i) the types T such that $t_i = 0$ increase the probability of not finding user i in the first round by (an additive factor of) $(m-1)(1+\varepsilon)\varepsilon\beta_i$.

To conclude (the two above arguments) the probability that the candidate solution finds i during the second round (and not during the first round) is at most $\beta_i(1+\varepsilon)+\beta_i(1+\varepsilon)(m-1)\varepsilon = \beta_i(1+\varepsilon)(1+(m-1)\varepsilon) \leq \beta_i(1+(m+1)\varepsilon)$ (since $\varepsilon < 1/(m+1)$).

We denote by $S = \{i \in U | \beta_i = \frac{\varepsilon}{n^2}\}$ the set of users with *small* probability of being left for the second round, and by $L = U \setminus S$ the set of users with *large* probability of being left for the second round.

Theorem 1. *The best candidate solution is a* $(1+\varepsilon)(1+2\varepsilon)(1+(m+1)\varepsilon)$*-approximated solution.*

Proof. We will analyze the candidate solution that corresponds to the right guesses (with respect to the information used by the solution OPT). By Lemma 5, this is a candidate solution. The best candidate solution clearly outperforms this particular candidate solution.

For a user $i \in L$, the probability that OPT does not find i in the first round is α_i, whereas by Lemma 6 the probability that the candidate solution does not find i in the first round is at most $\beta_i(1+(m+1)\varepsilon)$. Since $i \in L$, we conclude that $\alpha_i \geq \frac{\beta_i}{1+\varepsilon}$. Therefore, the probability that the candidate solution does not find i in the first round is at most $(1+\varepsilon)(1+(m+1)\varepsilon)\alpha_i$.

For a user $i \in S$, the probability that the candidate solution does not find i in the first round is at most $\beta_i(1+(m+1)\varepsilon) = \frac{\varepsilon(1+(m+1)\varepsilon)}{n^2} \leq \frac{2\varepsilon}{n^2}$, where the inequality follows as $\varepsilon < \frac{1}{(m+1)}$. Using the union bound we conclude that the probability that the candidate solution does not find at least one of the users in S is at most $\frac{2\varepsilon|S|}{n^2} \leq \frac{2\varepsilon m}{n^2} \leq \frac{2\varepsilon}{n}$, where the last inequality follows from the assumption $n > m$. In case this event happens we assign an extra cost of n (for the second round). This extra cost incurs an expected extra cost (an additive factor) of at most $\frac{2\varepsilon}{n} \cdot n = 2\varepsilon$. Since OPT costs at least 1, we will conclude that the users in S caused an increase of the approximation factor by a multiplicative factor of at most $1+2\varepsilon$.

We first assume that there is $\ell \in U$ such that $\beta_\ell(1+(m+1)\varepsilon)(1+\varepsilon) \geq 1$. In this case $\alpha_\ell \geq \frac{1}{(1+(m+1)\varepsilon)(1+\varepsilon)^2}$, and therefore $COST(OPT) \geq OPT_1 + (n-OPT_1)\alpha_\ell \geq n\alpha_\ell \geq \frac{n}{(1+(m+1)\varepsilon)(1+\varepsilon)^2} \geq \frac{n}{(1+(m+1)\varepsilon)(1+\varepsilon)(1+2\varepsilon)}$. Note that the returned solution costs at most n, and therefore in this case the returned solution pays at most $(1+(m+1)\varepsilon)(1+\varepsilon)(1+2\varepsilon)COST(OPT)$. Therefore, we can assume that for all ℓ, $\beta_\ell(1+(m+1)\varepsilon)(1+\varepsilon) < 1$.

We denote by $\tau = (1+\varepsilon)(1+(m+1)\varepsilon)$. The cost of the candidate solution is at most:

$$
\begin{aligned}
& OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in U} (1 - \beta_\ell (1 + (m+1)\varepsilon))\right) && (1)\\
& \le OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in L} (1 - \beta_\ell (1 + (m+1)\varepsilon))\right) + 2\varepsilon && (2)\\
& \le OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in L} (1 - \tau \alpha_\ell)\right) + 2\varepsilon && (3)\\
& \le OPT_1 + (n - OPT_1)\tau \cdot \left(1 - \prod_{\ell \in L} (1 - \alpha_\ell)\right) + 2\varepsilon && (4)\\
& \le (\tau + 2\varepsilon) \left[OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in L} (1 - \alpha_\ell)\right)\right] && (5)\\
& \le \tau(1 + 2\varepsilon) \left[OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in L} (1 - \alpha_\ell)\right)\right] && (6)\\
& \le \tau(1 + 2\varepsilon) \left[OPT_1 + (n - OPT_1) \cdot \left(1 - \prod_{\ell \in U} (1 - \alpha_\ell)\right)\right] && (7)\\
& = \tau(1 + 2\varepsilon) COST(OPT), && (8)
\end{aligned}
$$

where (1) follows from Lemma 6 and the monotonicity of the goal function (increasing the probability of not finding a user in the first round only increases the solution cost). (2) follows as explained above since the users in S incur an additive increase of the expected cost by at most 2ε. (3) follows since for all $i \in L$, $\alpha_\ell(1+\varepsilon) \ge \beta_\ell$. (4) follows because given a set of $|L|$ independent random events the probability of their union is multiplied by at most τ if we multiply the probability of each event in this set by τ. (5) and (6) follow by simple algebra (and by $OPT_1 \ge 1$). (7) follows since we deal with probabilities, and for each $\ell \in S$, $1 - \alpha_\ell \le 1$, and therefore $\prod_{\ell \in L}(1-\alpha_\ell) \ge \prod_{\ell \in U}(1-\alpha_\ell)$. (8) follows from the fact that we consider the right guesses on OPT.

By Theorem 1 we conclude that,

Corollary 1. *The above scheme is a* $[1 + 6m\varepsilon]$*-approximation for all* $\varepsilon > 0$.

Proof. Since $\varepsilon < \frac{1}{(m+1)}$ and $m \ge 2$ we get $(1 + (m+1)\varepsilon)(1+\varepsilon)(1+2\varepsilon) \le (1+(m+1)\varepsilon)(1+4\varepsilon) \le 1 + (m+9)\varepsilon \le 1 + 6m\varepsilon$.

By setting $\varepsilon' = \frac{\varepsilon}{6m}$, and applying the above algorithm with ε' instead of ε, we get a $1+\varepsilon$-approximation algorithm whose time complexity is polynomial for any fixed value of ε. Therefore, we proved the main result:

Theorem 2. *Problem EWCC with two rounds and a constant number of users has a polynomial time approximation scheme.*

4 Extension of the PTAS to Any Fixed Number of Rounds

In this section we show how the PTAS of the previous sections can be extended to provide a PTAS for EWCC when the number of rounds d is an arbitrary constant (the number of users is also a constant).

We fix an optimal solution OPT. Our scheme is again composed of two guessing steps.

Given OPT, denote by OPT_r the number of cells that OPT pages in the r-th round, and by α_i^r the probability for OPT to find user i exactly in the r-th round (i.e., OPT does not find i in the first $r-1$ rounds but finds i in the r-th round). Denote by $\pi_i^r = \sum_{s=r}^{d} \alpha_i^s$ the probability that OPT does not find i in the first $r-1$ rounds. Therefore, the cost of OPT denoted by $COST(OPT)$ is $COST(OPT) = \sum_{r=1}^{d} OPT_r \cdot (1 - \prod_{i \in U}(1 - \pi_i^r))$.

Recall that m, d are constants, and let ε be a value such that $0 < \varepsilon < \frac{1}{(md+1)}$. If $n \leq md^2$, then EWCC can be solved in a constant time via exhaustive enumeration (as m and d are constants), therefore we assume that $n > md^2$. Similarly to the $d = 2$ case we denote the probability intervals $I_0 = (0, \frac{\varepsilon}{n^2}]$, and for $1 \leq i \leq \left\lceil \log_{1+\varepsilon}\left(\frac{n^2}{\varepsilon}\right)\right\rceil$, $I_i = \left(\frac{\varepsilon}{n^2}(1+\varepsilon)^{i-1}, \frac{\varepsilon}{n^2}(1+\varepsilon)^{i}\right]$.

Our first guessing step guesses for each $\ell \in U$ and $1 \leq r \leq d$, the index $i_r(\ell)$ such that $\alpha_\ell^r \in I_{i_r(\ell)}$. The following lemma is trivial:

Lemma 7. *The number of possibilities for the first guessing step is*

$$O\left(\left\lceil \log_{1+\varepsilon}\left(\frac{n^2}{\varepsilon}\right) + 2\right\rceil^{md}\right) .$$

Therefore, performing an exhaustive enumeration for this guessing step can be done in polynomial time. We continue to analyze the iteration of this step in which we guess the right values that correspond to OPT. For all $\ell \in U$, we denote *the guess of* α_ℓ^r by β_ℓ^r to be the upper bound of $I_{i_r(\ell)}$; $\beta_\ell^r = \frac{\varepsilon}{n^2}(1+\varepsilon)^{i_r(\ell)}$.

The next step is to scale up the probabilities. Similarly in the $d = 2$ case we define $q_i^j(r) = p_i^j/(\beta_i^r)$ to be the scaled probability for user i to be found in cell j in round r. The matrix of cell j is $Q^j = (q_i^j(r))_{1 \leq i \leq m, 1 \leq r \leq d}$. For every matrix, each component larger than 1 is replaced by ∞ as this probability means that such cells cannot be paged in the relevant round. We further assign a type to each cell in the following way.

Let $q_i^j(r)$ be a maximum real entry in Q^j (if all entries are ∞, we can skip the current guess as it cannot lead to a valid solution), then we assign a weight $w^j = q_i^j(r)$ to Q^j, and we define $\tilde{Q}^j = \left(q_\ell^j(r)/w^j\right)_{\ell \in U, 1 \leq r \leq d}$. Note that $Q^j = w^j \cdot \tilde{Q}^j$. We define a set of intervals $\mathcal{J}$ as follows: $J_0 = (0, \varepsilon]$, and for all $k \geq 1$, $J_k = (\varepsilon \cdot (1+\varepsilon)^{k-1}, \varepsilon \cdot (1+\varepsilon)^k]$, and $\mathcal{J} = \{J_0, J_1, \ldots\}$. For each entry (ℓ, r) in $\tilde{Q}^j$, we find the interval from $\mathcal{J}$ that contains $q_\ell^j(r)/w^j$. We assign the type of Q^j to be the following matrix. For each (ℓ, r) of real probability, compute a value $t_{(\ell,r)}$

such that $\frac{q_\ell^j(r)}{w^j} \in J_{t_{(\ell,r)}}$. Entries of infinite probability are assigned $(t_{(\ell,r)} = \infty$. The type of Q^j is the matrix $(t_{(\ell,r)})_{1\leq \ell\leq m, 1\leq r\leq d}$.

Our second guessing step is to guess OPT_r for $r = 1, 2, \ldots, d$ (since the first round is never skipped, OPT_1 is an integer between 1 and n, and the other values are integers between 0 and $n-1$) and guess the *number* of cells from each type that OPT pages in each round.

Lemma 8. *The number of possible types is* $O\left(\left[\log_{1+\varepsilon}\left(\frac{1}{\varepsilon}\right)+3\right]^{md}\right)$.

Proof. Each entry can have any of the values as in the two round case or infinity.

Lemma 9. *The number of possible guesses is bounded by the value*
$O\left((n+1)^{d\left[\log_{1+\varepsilon}\left(\frac{1}{\varepsilon}\right)+3\right]^{md}}\right)$.

Proof. For round $1 \leq r \leq d$, the number of cells from each type is an integer between 0 and $OPT_r \leq n$. Guessing the number of cells of each type in every round implies a guess of the OPT_r values.

Note that the number of possibilities for this guessing step is polynomial (for fixed values of ε, m, d).

Assume that for a type T, OPT has $OPT_r(T)$ cells of type T that are paged in the r-th round. We sort the cells of type T according to their weight, and we iterate the following: we initialize $r = d$ and assign the r-th round $OPT_r(T)$ cells (among the cells of type T) that have the least weight. We remove this set of cells, we decrease r by 1 and repeat until no more cells of type T exist. We apply this procedure for all the types T. We would like to ignore all invalid solutions. In order to be valid, the probability bounds β_ℓ^r must be satisfied, i.e. the sum of probabilities must be in the interval $I_{i_r(\ell)}$. We slightly relax this requirement since a result of the scaling may shift the sum out of this interval. Instead, we disregard probabilities such that their scaled probability is in the interval $(0, \varepsilon]$, and we require that the sum over all rounds from r to d, of the sum of the other (non-scaled) probabilities should reside in the interval $[0, \sum_{s=r}^{d} \beta_\ell^s(1+\varepsilon)]$. For some guesses we obtain a candidate solution. Among all candidate solutions we output the one whose cost is minimized.

Lemma 10. *For a fixed number of users m, a fixed number of rounds d, and for a constant $\varepsilon > 0$, the above scheme takes a polynomial time.*

Proof. By Lemmas 7,8 and 9, the number of possibilities in the first step and in the second step is polynomial. The time to compute the resulting candidate solution for a single guess is clearly polynomial (i.e., finding a maximum value for each cell and finding its weight is polynomial, and the rest is simply sorting of the cells according to their weights), and the time to compute its cost is also polynomial. Therefore, the scheme takes a polynomial time.

We analyze the iteration of the first guessing step in which the guessed values of β_i^r $\forall i, r$ are the right guesses. We also assume that in the second guessing step we guess the right values of OPT_r for $r = 1, 2, \ldots, d$ and the right number of cells of each type that OPT pages in each round. We analyze the cost of the corresponding candidate solution.

Lemma 11. *The right set of guesses leads to a candidate solution.*

Proof. We have to show that for each user i and each round r, the sum of the probabilities (when we ignore cells whose scaled probability is at most ε) of not finding user i within the first $r-1$ rounds is at most $\sum_{s=r}^{d} \beta_i^s(1+\varepsilon)$. For a type T, OPT selects $\sum_{s=r}^{d} OPT_s(T)$ cells of type T with sum of weights that is at least the sum of weights of the cells of type T that the candidate solution selects (note that the weights are not changed in the process of partitioning the cells into types). By definition of $\mathcal{J}$, the probabilities of having user i in a pair of cells of type T with the same weight, are within a multiplicative factor of $1+\varepsilon$. Therefore, the contribution of type T cells to the probability that the candidate solution does not find i during the first $r-1$ rounds, is at most $1+\varepsilon$ times the contribution of type T cells to the probability that OPT does not find i during the first $r-1$ rounds. As the probability that OPT finds i only during the s-th round is $\alpha_i^s \leq \beta_i^s$, the claim follows.

Lemma 12. *Consider a user i and a round r, then the probability that the candidate solution does not find i during the first $r-1$ rounds is at most $\sum_{s=r}^{d} \beta_i^s(1+(md+1)\varepsilon)$.*

Proof. Consider a type matrix T. A type with an ∞ entry for round s will have zero cells for that round. Otherwise assume first that $t_{(i,r)} \geq 1$. By Lemma 11, the contribution of type T cells to the probability that the candidate solution does not find i during the first $r-1$ rounds, is at most $1+\varepsilon$ times the contribution of type T cells to the probability that OPT does not find i during the first $r-1$ rounds.

Next, consider a type T such that $t_{(i,r)} = 0$. For such a type we define the *leader* of T to be the first entry of the type matrix that relates to the largest real interval (that contains the point 1). There exists at least one entry like this, as there is at least one unit entry in $\tilde{Q}^j$. Note that the sum of scaled probabilities of finding user ℓ in round r' of all the cells paged by OPT in that round with a type such that ℓ is the leader, is at most $1+\varepsilon$. Therefore, the total contribution of scaled probabilities of all the cells of any type T such that $t_{(i,r')} = 0$ and ℓ, r' is acting as the leader of T is at most $(1+\varepsilon)\varepsilon$. Summing over all ℓ and r' (note that we may exclude the case $\ell = i, r' = r$), we get an increase of $(md-1)\varepsilon(1+\varepsilon)$ caused by the types where $t_{(i,r)} = 0$. In terms of the original probabilities (i.e., for each cell and round s we multiply its probability by β_i^s) the types T such that $t_{(i,r)} = 0$ increase the probability of not finding user i in the first $r-1$ rounds by at most (an additive factor of) $(dm-1)\varepsilon(1+\varepsilon)\sum_{s=r}^{d} \beta_i^s$.

To conclude (the two above arguments) the probability that the candidate solution does not find i during the first $r-1$ rounds is at most $\sum_{s=r}^{d} \beta_i^s(1+\varepsilon) + \sum_{s=r}^{d} \beta_i^s(md-1)\varepsilon(1+\varepsilon) = \sum_{s=r}^{d} \beta_i^s(1+(md+1)\varepsilon)$ (since $\varepsilon < 1/(dm+1)$).

.

Theorem 3. *The best candidate solution is a* $(1+\varepsilon)^2(1+(md+1)\varepsilon)$*-approximated solution.*

Proof. We will analyze the candidate solution that corresponds to the right guesses (with respect to the information used by the solution OPT). By Lemma 11, this is a candidate solution. The best candidate solution clearly outperforms this particular candidate solution.

For a user i, the probability that OPT does not find i in the first $r-1$ rounds is $\sum_{s=r}^{d} \alpha_i^s$, whereas by Lemma 12 the probability that the candidate solution does not find i in the first $r-1$ rounds is at most $\sum_{s=r}^{d} \beta_i^s(1+(md+1)\varepsilon)$. For all $\ell \in U$ and for all $s = 1, 2, \ldots, d$, $\alpha_\ell^s(1+\varepsilon) + \frac{\varepsilon}{n^2} \geq \beta_\ell^s$ holds. This gives $\sum_{s=r}^{d} \alpha_i^s \geq \frac{\sum_{s=r}^{d} \beta_i^s}{1+\varepsilon} - \frac{\varepsilon(d-r+1)}{n^2}$. Therefore, the probability that the candidate solution does not find i in the first $r-1$ rounds is at most $(1+\varepsilon)(1+(md+1)\varepsilon)\sum_{s=r}^{d} \alpha_i^s + (1+\varepsilon)(1+(md+1)\varepsilon)\frac{\varepsilon d}{n^2}$. Since the above term bounds a probability we conclude that the probability that the candidate solution does not find i in the first $r-1$ rounds is at most $\min\{1, (1+\varepsilon)(1+(md+1)\varepsilon)\sum_{s=r}^{d} \alpha_i^s + (1+\varepsilon)(1+(md+1)\varepsilon)\frac{\varepsilon d}{n^2}\}$.

We denote by $\tau = (1+\varepsilon)(1+md\varepsilon)$. The cost of the candidate solution is at most:

$$OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \left(1 - \prod_{\ell \in U}\left(1 - \sum_{s=r}^{d} \beta_\ell^s(1+md\varepsilon)\right)^+\right) \tag{9}$$

$$\leq OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \left(1 - \prod_{\ell \in U}\left(1 - \tau\left(\sum_{s=r}^{d} \alpha_\ell^s + \frac{\varepsilon d}{n^2}\right)\right)^+\right) \tag{10}$$

$$\leq OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \left(1 - \prod_{\ell \in U}\left(1 - \tau\sum_{s=r}^{d} \alpha_\ell^s\right)^+ + \frac{\varepsilon\tau md}{n^2}\right) \tag{11}$$

$$\leq OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \left(1 - \prod_{\ell \in U}\left(1 - \tau\sum_{s=r}^{d} \alpha_\ell^s\right)^+\right) + \frac{\varepsilon\tau md^2}{n} \tag{12}$$

$$\leq OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \left(1 - \prod_{\ell \in U}\left(1 - \tau\sum_{s=r}^{d} \alpha_\ell^s\right)^+\right) + \varepsilon\tau \tag{13}$$

$$\leq OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \tau\left(1 - \prod_{\ell \in U}\left(1 - \sum_{s=r}^{d} \alpha_\ell^s\right)\right) + \varepsilon\tau \tag{14}$$

$$\leq \tau(1+\varepsilon)\left[OPT_1 + \sum_{r=2}^{d} OPT_r \cdot \left(1 - \prod_{\ell \in U}\left(1 - \sum_{s=r}^{d} \alpha_\ell^s\right)\right)\right] \tag{15}$$

$$= \tau(1+\varepsilon)COST(OPT), \tag{16}$$

where (9) follows from Lemma 12 and the monotonicity of the goal function (increasing the probability of not finding a user in the first rounds only increase the solution cost). (10) follows as explained above. (11) follows by simple algebra. (12) follows since $OPT_r \leq n$, $\forall n$. (13) follows from the assumption $n \geq md^2$. (14) follows because given a set of $|L|$ independent random events the probability of their union is multiplied by at most τ if we multiply the probability of each event in this set by τ. (15) follow by simple algebra (and by $OPT_1 \geq 1$). (16) follows from considering the right guesses on OPT.

Similar to the $d = 2$ case, we establish the following theorem:

Theorem 4. *Problem EWCC with a constant number of rounds and a constant number of users has a polynomial time approximation scheme.*

References

1. A. Bar-Noy and G. Malewicz, "Establishing wireless conference calls under delay constraints," *Journal of Algorithms*, **51**, 145-169, 2004.
2. D. Goodman, P. Krishnan and B. Sugla, "Minimizing queuing delays and number of messages in mobile phone location," *Mobile Networks and Applications*, **1**, 39-48, 1996.
3. C. Rose and R. Yates, "Minimizing the average cost of paging under delay constraints," *Wireless Networks*, **1**, 211-219, 1995.