Establishing Wireless Conference Calls Under Delay Constraints *

Amotz Bar-Noy †

amotz@sci.brooklyn.cuny.edu

Grzegorz Malewicz ‡

greg@cse.uconn.edu

June 23, 2002

Abstract

A prevailing feature of mobile telephony systems is that the cell where a mobile user is located 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 are located in, to find the cells where the users currently reside. The search consumes expensive wireless links and so it is desirable to develop search techniques that page as few cells as possible.

We consider cellular systems with c cells and m mobile users roaming among the cells. The location of the users is uncertain as given by probability distribution vectors. Whenever the

*A preliminary version of this work appeared in the Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002). Address for correspondence: Grzegorz Malewicz, Department of Computer Science and Engineering, 191 Auditorium Rd., Unit 155, University of Connecticut, Storrs, CT 06269, USA. E-mail: greg@cse.uconn.edu, tel (860) 486-5570, fax (860) 486-4817.

†Computer and Information Science Department, CUNY-Brooklyn College, New York. The work of the first author was done during his visit at the AT&T Research Labs, Florham Park, NJ.

‡Department of Computer Science and Engineering, 191 Auditorium Rd., Unit 155, University of Connecticut, Storrs, CT 06269, USA. The work of the second author was done during his summer internship at AT&T Research Labs, Florham Park, NJ.

system needs to find specific users, it conducts a search operation lasting some number of rounds (the delay constraint). In each round the system may check an arbitrary subset of cells to see which users are located there. In this setting the problem of finding one user with minimum expected number of cells searched is known to be solved optimally in polynomial time.

In this paper we address the problem of finding several users with the same optimization goal. This task is motivated by the problem of establishing a conference call between mobile users. We first show that the problem is NP-hard. Then we prove that a natural and simple heuristic is a $\frac{e}{e-1}$ approximation solution.

Key words: Location management, conference call, terminal paging, NP-hardness, approximation algorithms, convex optimization.

1 Introduction

In the last decade, we have witnessed two trends: increasing availability of people and increasing availability of information. Mobile telephony systems make it possible to talk with people even if they are not residing in predetermined locations (as is the case with conventional phone systems). Internet search engines allow users to accurately and efficiently access information stored on websites that have fixed location. When information is stored on mobile devices and needs to be retrieved new challenges occur. An intrinsic feature of current mobile telephony systems is that location of the devices is uncertain. A challenge here is to design search algorithms that efficiently retrieve information given limited knowledge about its location.

In this paper we focus on search techniques that are motivated by the problem of establishing a conference call in a wireless phone system. Our goal is to find a given collection of mobile devices inside a wireless system so as to minimize the usage of expensive wireless links and control the amount of time spent on the search.

1.1 Background and motivation

Our problem of establishing wireless conference calls is motivated by the state-of-the-art of wireless technology. Currently deployed wireless personal communication systems are composed of a set of *base stations* connected by a wired *backbone network*. The range of radio transmission of a base station determines an area called a *cell*. Each *mobile device* that roams inside the system communicates with other devices (mobile or stationary) through base stations using radio signals. When a mobile device is within the range of a base station the terminal is said to be *located in the cell* determined by the base station.

One of the main components of a wireless system is a *location management* service [2]. Its goal is to track the locations of devices that are needed in order to establish calls. In some cases the system knows the current location of a mobile device and then the system does not need to search for the device to establish a call. This occurs, for example, when the device is participating in an ongoing call and therefore is repeatedly communicating with base stations. However in general the location of a device may be unknown for example because the device has been switched off by the user in order to preserve battery power. This gives rise to the need of locating mobile devices which is the aim of a location management service.

The tracking problem exhibits an inherent tradeoff between the load imposed on the system by devices reporting their location and the load imposed when the system searches for devices (see for example [5]). To illustrate, let us assume that each terminal *reports* its location by sending a message over a wireless link to the base station every time it enters a new cell. This means that the system has up-to-date information about terminal location and when call is to be established the system does not need to search for a device. Assume on the other hand that terminals never report their location. Then when the system is to establish a call with a terminal it must find the cell where the terminal is currently located. Since the terminal is mobile the system may need to search all cells in the system, which is done by broadcasting over wireless links, or *paging*, from base stations.

Several factors affect the total cost incurred by the system due to reporting and paging. Reporting location consumes limited battery power of terminals. In addition the wireless links used for reporting are expensive and may become congested by the volume of update traffic especially when terminals are highly mobile and frequently cross the boundaries between cells. Paging also may consume scarce wireless links specifically when large number of cells needs to be paged to find the device and the incoming call frequency is large. An effective and efficient location management system needs to achieve a balance between reporting and paging depending on these factors.

Presently major wireless systems use a simple technique to balance between reporting and paging traffic. In GSM MAP [11] (used in Europe) and IS-41 [10] (used in North America) standards the set of all cells is partitioned into subsets called *location areas* each containing many cells (but only a fraction of all cells in the system). Any cell broadcasts the identifier of its location area on a special radio channel. When a mobile terminal finds out that it has moved from a cell to a cell belonging to a different location area it sends a wireless signal (reports) to the base station of the later cell. This information is persisted in a database connected to the backbone network, that stores the most recently visited location area for each terminal. When a call to a mobile terminal is to be established the system broadcasts (pages) radio signals in parallel from all base stations in the location area asking the terminal to respond. This technique reduces the number of cells paged when a call is to be established because only the cells belonging to a location area are paged. However this comes at the cost that devices must report whenever they cross boundaries between location areas. The choice of location areas affects the reporting traffic (e.g., [1, 6]).

A technique has been developed [13, 18, 19] to reduce the number of cells paged inside a location area during the search for a single mobile device at the cost of increased amount of time needed for paging. The technique considers a model of a location area with c cells where the probability distribution of the device across the cells is given. (There are several methods for approximating the distribution; see [17, 18] for examples). Here an arbitrary subset of cells of the location area can be paged in parallel in unit time to see if the device is located there. Also paging can be performed for $d \leq c$ units of time. Paging is carried out according to a d-round strategy, where in each round i a subset $S_i \subseteq [c] = \{1, \ldots, c\}$ of cells is paged until the device is found. It is assumed that the device does not move during the search. Authors show how to efficiently find a strategy that has at most d rounds and that minimizes expected number of cells paged. For example if the device is uniformly distributed across all c cells, c even, and we have at most 2 units of time for paging then the best paging strategy is to page half of the cells in the first round and when the device is not found in these cells then page the other half in the second round. This gives $\frac{3}{4}c$ expected number of cells paged (a $\frac{1}{4}c$ improvement over the technique specified by GSM MAP and IS-41 standards).

It is very interesting to investigate how to generalize the technique to find more than one device. This is motivated by the problem of establishing a wireless conference call. The goal is to find a given collection of mobile devices inside a wireless system so as to minimize the usage of expensive wireless links and control the amount of time spent on the search.

1.2 Model and problem statement

Assume a model with c cells and m mobile devices. We assign all the devices to cells by selecting a cell j for a device i with probability $p_{i,j}$ independently from other devices. We assume that each probability is positive and that $p_{i,1} + \ldots + p_{i,c} = 1$, for all i. In this setting we can probe (or page) an arbitrary group G of cells in a unit of time and detect, for each device located in one of the cells from G, the cell where the device is located. We also assume that we can afford to perform probing for d units of time.

The goal is to develop efficient algorithms that can find all the devices within at most d units of time and that page the least expected number of cells. In general one can consider two types of algorithms: *oblivious* and *adaptive*. Oblivious algorithms page, in each unit of time, a predetermined subset of cells. Adaptive algorithms decide which cells to page during a unit of time based on the devices that have been found so far. Each type has its advantages. Adaptive algorithms may achieve lower expected number of cells paged to find all terminals. Oblivious algorithms have very low computational cost during the search process. In this work we focus on oblivious algorithms.

Specifically, a *strategy* is a sequence $\langle S_1, \ldots, S_t \rangle$ of nonempty sets that partition $[c] = \{1, \ldots, c\}$, where t is called the *length* of the strategy. We call each set a *group*. For a given strategy cells are paged in *rounds* such that in round r, $1 \leq r \leq t$, all cells in the group S_r are paged. Cells in $S_{r+1}, \ldots, S_t$ are not paged when all mobile devices have been found i.e., if and only if r is the smallest round number for which all mobile devices are located in cells $S_1 \cup \ldots \cup S_r$. For each strategy we can find the expected number of cells paged until all mobile devices have been found. We call this number the *expected paging* of the strategy. We seek a strategy that minimizes this expectation. We call the problem the Conference Call problem.

Conference Call

Instance: Number of mobile devices $m \geq 1$, number of cells $c \geq 1$, positive rational probabilities $p_{i,j}$ of finding mobile device i in cell j such that for all i, $p_{i,1} + \ldots + p_{i,c} = 1$, and a maximum number of rounds d, $1 \leq d \leq c$.

Objective: Find a strategy that minimizes expected paging for all strategies of length at most d.

For example when $d = 1$, the problem is trivial since all the cells must be paged in the first and only round. The problem becomes interesting when $d = 2$, because we are looking for a subset of the cells to be paged in the first round to minimize the expected number of cells paged until all devices have been found (if they are not located in the cells paged in the first round then we need to page all the remaining cells). When $d = c$, we are looking for a permutation of the cells that dictates the sequence in which cells are paged that minimizes the expected number of cells paged until all devices have been found.

1.3 Contributions

The problem of searching for one device ($m = 1$) is known [13, 18, 19] to be solvable optimally for any d in polynomial time using a dynamic programming algorithm. In this paper we address the general problem of searching for $m \geq 1$ devices. Our contributions are as follows:

Complexity of the problem. We first show that the Conference Call problem is NP-hard by a nontrivial reduction from a special type of the Partition problem. We also show that the special case of the Conference Call problem for constant number of devices $m \geq 2$ and constant number of rounds $d \geq 2$ is NP-hard. This appears in Theorem 3.8. Our result establishes a threshold: the problem is easy when $m = 1$ or when $d = 1$, but becomes difficult when $m = 2$ and $d = 2$.

Constant factor approximation. We present a very simple and natural heuristic. In this heuristic we sequence the cells in a non-increasing order of expected number of devices located in each cell and use dynamic programming to find the best partition of this sequence into d subsequences. We show that the expected number of cells paged by this heuristic is at most $\frac{e}{e-1}$ times the minimal number of cells paged by any strategy. This appears in Theorem 4.8. We also show a lower bound of $\frac{320}{317}$ on the performance ratio of our heuristic.

1.4 Paper organization

The rest of the paper is structured as follows. In Section 2, we present some preliminaries. In Section 3, we prove that the problem is NP-hard and that special cases of the problem are also NP-hard. In Section 4 we show a constant factor approximation for the problem. Finally, in Section 5, we discuss future work and related work.

2 Preliminaries

We show that the Conference Call problem is a combinatorial optimization problem and restate some existing results from convex optimization that we use throughout. In the lemma below we show how to find the expected paging for a strategy.

Lemma 2.1. *Let $\langle S_1, \ldots, S_t \rangle$ be a strategy. Then the expected number of cells paged until all mobile devices are found is*

$$EP = c - \sum_{r=1}^{t-1} |S_{r+1}| \prod_{i=1}^{m} \sum_{j \in L_r} p_{i,j} \ ,$$

where $L_r = S_1 \cup \ldots \cup S_r$.

Proof. The search lasts exactly r rounds whenever not all mobile devices are found in rounds $1, \ldots, r-1$, but all are found on or before round r. Let F_r be the event that all mobile devices are found on or before round r. Observe that by independence $\Pr[F_r] = \prod_{i=1}^{m} \sum_{j \in L_r} p_{i,j}$. Hence

$$\Pr[\text{paging lasts exactly } r \text{ rounds}] = \Pr[F_r] - \Pr[F_{r-1}] \ .$$

Since if paging stops in round r we page $|S_1| + \ldots + |S_r|$ cells, the expected number of cells paged until all mobile devices are found is

$$EP = \sum_{r=1}^{t} (|S_1| + \ldots + |S_r|) (\Pr[F_r] - \Pr[F_{r-1}]) = c \Pr[F_t] - \sum_{r=1}^{t} |S_r| \Pr[F_{r-1}] \ ,$$

which completes the proof. □

It can be seen that for any strategy of length $t-1<c$ there exists a strategy of length t with strictly lower expected paging. Thus, among all strategies of length at most d, a strategy that minimizes expected paging must have length d.

Now we restate some basic notions and properties from the theory of multidimensional convex optimization [20] and linear algebra [16] that we use throughout. A subset D of space $\mathbb{R}^k$ is *convex* if for any two points x and y in D the segment xy is contained in D. That is $\lambda x + (1-\lambda)\, y \in D$, for all $x, y \in D$, $\lambda \in [0,1]$. For a convex set D, a function $f : D \to \mathbb{R}$ is *strictly convex* if for any x and y in D and $\lambda \in (0,1)$ we have $f\left(x\lambda + (1-\lambda)\, y\right) < f\left(x\right)\lambda + f\left(y\right)(1-\lambda)$. A matrix H of size k by k over $\mathbb{R}$ is *positive definite* if for any $x \in \mathbb{R}^k$, $x^T H x > 0$, unless $x = 0$. We report a standard fact that characterizes strict convexity of a function in terms of its Hessian – the matrix of second derivatives (see [20] for a proof and more discussion).

Theorem 2.2 ([20]). *Let D be an open convex set in $\mathbb{R}^k$. Suppose that a differentiable function $f : D \to \mathbb{R}$ has continuous second partial derivatives in the set D. Let the Hessian $H(x)$ be positive definite for every point $x \in D$. Then f is strictly convex.*

Using the above theorem it is strightforward to show that the maximum of a strictly convex function are attained at the boundary of the domain as presented in the next lemma.

Lemma 2.3. *Let $f : D \to \mathbb{R}$ be a strictly convex and continuous function defined over an open, convex and bounded subset $D \subset \mathbb{R}^k$, and let H be a closed subset of D. Then the maximum of f on H is achieved at a point that belongs to the boundary of H.*

3 Complexity of the problem

This section is dedicated to showing that the Conference Call problem is NP-hard. We demonstrate that it contains a subproblem that is NP-hard. This subproblem is the Conference Call problem restricted to $m = 2$ and $d = 2$. Our result establishes a threshold because the Conference Call problem is in P for $m = 1$ [13, 18, 19] or for $d = 1$. In our proof we reduce a certain variation of the Partition problem to the restricted Conference Call problem. Next we generalize the techniques used in the proof and show that any restriction of the Conference Call problem to constant $m \geq 2$ and $d \geq 2$ is also NP-hard.

3.1 The problem is hard

We show that the Conference Call problem is NP-hard by a transformation from the following Partition problem which is NP-complete as presented by Garey and Johnson [12] on page 223.

Partition

Instance: A number g divisible by 2 and positive integer sizes $s_1, \ldots, s_g$.

Objective: Decide if there is a subset P of $[g]$ such that $|P| = g/2$ and $\sum_{k \in P} s_k = \frac{1}{2} \sum s_k$.

The Partition problem can be reduced to the following variation of the Partition problem called Quasipartition1. We omit the proof because Quasipartition1 is one of problems from the family Quasipartition2 defined in the following section, and there we show that any of these problems is NP-complete.

Quasipartition1

Instance: A list of non-negative rational sizes $s_1, \ldots, s_c$, where c is divisible by 3.

Objective: Decide if there is a subset I of $[c]$ such that $|I| = \frac{2}{3}c$ and $\sum_{k \in I} s_k = \frac{1}{2} \sum s_k$.

We use the Quasipartition1 problem to show NP-hardness of the Conference Call problem. The following technical lemma that analyzes extrema of a function is used in the proof.

Lemma 3.1. *For any $c \geq 1$ the function $f : [0,1] \times [0,c]$ defined as*

$$f(x,y) = (c-y)\left(\left(1 - \frac{3}{2c}\right)y + x\right)(y-x)$$

achieves the global maximum at a single point of the domain $x = 1/2$ and $y = \frac{2}{3}c$.

Proof. Let us extend the domain of f to the set $D = \{(x, y) : 0 \leq x \leq y + 1 \text{ and } 0 \leq y \leq c\}$. The function f is continuous over closed and bounded domain D so it achieves maxima either in the interior or at the boundary of D.

Let us focus on the interior first. Observe that $\frac{\partial f}{\partial x} = 0$ if and only if $x = \frac{3y}{4c}$, so an extremum in the interior can be achieved only when this condition holds. The first derivative of the function $f(\frac{3y}{4c}, y)$ is zero only when $y = \frac{2}{3}c$, or $y = 0$. Thus if there is extremum in the interior it is for $x = 1/2$, and $y = \frac{2}{3}c$, where the function achieves the value $f(\frac{1}{2}, \frac{2}{3}c) = \frac{4}{27}c^3 - \frac{2}{9}c^2 + \frac{1}{12}c$.

Now let us consider the values of the function at the boundary. If $y = c$ or $y = 0$ then $f(x, y) \leq 0$ for $x \geq 0$. If $x = 0$ then $\frac{\partial f(0,y)}{\partial y}$ is zero only when $y = 0$ or $y = \frac{2}{3}c$. Since $f(0, 0) = 0$, $f(0, \frac{2}{3}c) = \frac{4}{27}c^3 - \frac{2}{9}c^2$, and $f(0, c) = 0$, for the part of boundary where $x = 0$ the function achieves values strictly smaller than $f(\frac{1}{2}, \frac{2}{3}c)$. Finally if $x = y + 1$, $\frac{\partial^2 f(y+1,y)}{\partial y^2} = 4 - \frac{3}{c} > 0$, and so the maximum is achieved when $y = 0$ or when $y = c$, in which cases the function has non-positive values $-c$ and 0 respectively.

Thus the function f defined for a smaller domain $0 \leq x \leq 1$, $0 \leq y \leq c$ achieves the global maximum at a single point $x = 1/2$, and $y = \frac{2}{3}c$. □

Lemma 3.2. *The Conference Call problem restricted to $m = 2$ and $d = 2$ is NP-hard.*

Proof. We show how to transform the Quasipartition1 problem to the restricted Conference Call problem.

Let $s_1, \ldots, s_c$ be any list of non-negative rational sizes such that c is divisible by 3, and let $S = s_1 + \ldots + s_c$. If there exists i such that $s_i = S$, then there is no partition. For the remainder of the proof assume that $s_i < S$, for all i. We define an instance of the Conference Call problem restricted to $d = 2$ and $m = 2$. Since there are only two mobile devices in this instance for clarity we denote the probabilities that mobile devices are located in cell i by p_i and q_i respectively. Let the probabilities p_i and q_i be defined as $p_i = \frac{1}{c-1/2}\left(1 - \frac{3}{2c} + \frac{s_i}{S}\right)$, $q_i = \frac{1}{c-1}\left(1 - \frac{s_i}{S}\right)$, for $1 \le i \le c$. Note that the probabilities are positive, that $\sum p_i = 1$ and $\sum q_i = 1$, and that the size of the instance of the Conference Call problem is polynomial in the size of the instance of the Quasipartition1 problem. We show that the Quasipartition1 problem for the sequence $s_1, \ldots, s_c$ has an answer if and only if we can answer whether the minimal expected paging for the probabilities achieve a certain value defined below.

Before we proceed with the proof, we note a lower bound on the value of expected paging for the instance $m = 2$, $d = 2$, $p_1, \ldots, p_c$, $q_1, \ldots, q_c$. Take any strategy and let $I \subseteq [c]$, $|I| = y$, be the set of cells paged in the first round. By Lemma 2.1 the expected paging for this strategy is

$$\begin{aligned} EP &= c - (c - |I|)\left(\sum_{j\in I} p_j\right)\left(\sum_{j\in I} q_j\right) \\ &= c - \frac{c-y}{(c-1/2)(c-1)}\left(\left(1 - \frac{3}{2c}\right)y + \sum_{j\in I}\frac{s_j}{S}\right)\left(y - \sum_{j\in I}\frac{s_j}{S}\right) . \end{aligned}$$

Let us denote $\sum_{j\in I}\frac{s_j}{S}$ by x, and observe that $0 \le x \le 1$. Consequently by Lemma 3.1 the expected number of cells paged until all mobile devices are found is bounded from below by

$$EP \ge LB = c - \frac{1}{(c-1/2)(c-1)} \cdot f\left(\frac{1}{2}, \frac{2}{3}c\right)$$

Suppose that P is a partition of cardinality $\frac{2}{3}c$ of the sequence $s_1, \ldots, s_c$. Then $\sum_{j\in P}\frac{s_j}{S} = 1/2$, and so a strategy that pages cells P in the first round, and cells $[c] \setminus P$ in the second round has expected paging equal to LB. Moreover, by Lemma 3.1, this is the smallest value possible, so the answer to the instance of the Conference Call is a strategy with expected paging equal to LB.

Suppose that the answer to the instance of the Conference Call problem is a strategy with expected paging equal to LB. Let P be the set of cells paged in the first round. Observe that the cardinality of P must be $\frac{2}{3}c$ because otherwise by Lemma 3.1 the expected paging of the strategy would be strictly grater than LB. For the same reasons $\sum_{j\in P}\frac{s_j}{S}$ must be $1/2$. Thus P is a partition of cardinality $\frac{2}{3}c$ of the sequence $s_1,\ldots,s_c$. □

Corollary 3.3. *The Conference Call problem is NP-hard.*

3.2 Restrictions are also hard

It is interesting to consider restrictions of the Conference Call problem to constant d and m because in specific applications one may wish to have an algorithm that constructs paging strategies for some fixed number of mobile devices and some fixed maximum delay but for any location area (so that c and the probabilities are the input to the algorithm). It turns out that such restrictions are also NP-hard and we show it in this section by generalizing techniques from the previous section.

Let us begin with a technical lemma that generalizes Lemma 3.1. The lemma below will be used to show that the expected paging is minimized when the sets paged in each round have specific cardinalities.

Lemma 3.4. *Let $2\le d\le c$, and $i_1,\ldots,i_d$ satisfy constraints $i_j\ge 0$, $i_1+\ldots+i_d=c$, and $x_1,\ldots,x_d$ satisfy constraints $x_i\ge 0$, $x_1+\ldots+x_d=1$. Let $m\ge 2$, and $\alpha_1=\frac{m}{m+1}$, $\alpha_k=\frac{m}{m+1-\alpha_{k-1}^m}$, for $2\le k\le d-1$, $b_d=c$, $b_{k-1}=\alpha_{k-1}b_k$, for $2\le k\le d$, $b_0=0$. Then the function*

$$f(i_1,\ldots,i_d,x_1,\ldots,x_d)$$
$$=c-\frac{1}{c(c-1)}\sum_{r=1}^{d-1} i_{d+1}\left(\left(1-\frac{1}{c}\right)\sum_{k=1}^{r} i_k+\sum_{k=1}^{r} x_k\right)\left(\sum_{k=1}^{r} i_k-\sum_{k=1}^{r} x_k\right)\left(\frac{\sum_{k=1}^{r} i_k}{c}\right)^{m-2}$$

is strictly greater than

$$c-\frac{(2c-1)^2}{4\,(c-1)\,c^{m+1}}\sum_{r=1}^{d-1}\left(b_{r+1}-b_r\right)b_r^m\ ,$$

with equality only when $i_j=b_j-b_{j-1}$, for $1\le j\le d$, and $x_j=\frac{b_j}{2c}$, for $1\le j\le d-1$ and $x_d=1-\sum_{k=1}^{d-1}x_k$.

Proof. Let $b_r=\sum_{k=1}^{r} i_k$. Then by the theorem of arithmetic and geometric means we have

$$\left(\left(1-\tfrac{1}{c}\right)\sum_{k=1}^{r} i_k+\sum_{k=1}^{r} x_k\right)\left(\sum_{k=1}^{r} i_k-\sum_{k=1}^{r} x_k\right)\leq\left(\frac{\left(2-\frac{1}{c}\right)b_r}{2}\right)^2 ,$$

and so

$$f>c-\tfrac{1}{c(c-1)}\sum_{r=1}^{d-1}\left(b_{r+1}-b_r\right)\left(1-\tfrac{1}{2c}\right)^2\tfrac{b_r^m}{c^{m-2}}=c-\tfrac{(2c-1)^2}{4(c-1)c^{m+1}}\sum_{r=1}^{d-1}\left(b_{r+1}-b_r\right)b_r^m ,$$

with equality only when $\sum_{k=1}^{r} x_k=\frac{b_r}{2c}$, for all $1\leq r\leq d-1$.

Our goal now is to find an upper bound on the sum $\sum_{r=1}^{d-1}\left(b_{r+1}-b_r\right)b_r^m$ under constraints $0\leq b_1\leq\ldots\leq b_d=c$. Sine the domain for $b_1,\ldots,b_d$ is bounded and closed, and $\sum_{r=1}^{d-1}\left(b_{r+1}-b_r\right)b_r^m$ is a continuous function of b_i's, it achieves the maximal value at a point of the domain. We first show that the maximum must be achieved in the interior of the domain.

Let us take a point p from the boundary and use two actions to find a point in the interior with strictly larger sum. (Action 1) Let k be the largest index for which $b_k=0$ (so that $k<d$). We can make the sum strictly larger by taking $b_k=\frac{b_{k+1}}{2}$. After applying this action at most d times we have that $b_i>0$ for all i. (Action 2) Let k be the smallest index for which $b_k=b_{k+1}$. If $k=1$ then $\frac{\partial}{\partial b_1}=-b_1^m+mb_1^{m-1}\left(b_2-b_1\right)<0$ and so we can make the sum larger by slightly reducing b_1. If $1<k<d$ then $\frac{\partial}{\partial b_k}=b_{k-1}^m-b_k^m+mb_k^{m-1}\left(b_{k+1}-b_k\right)<0$, because $b_{k-1}<b_k$. Again we can increase the sum by slightly reducing b_k. After applying this action at most d times we obtain a point from the interior that has the value of the sum strictly greater than the value of the sum for the boundary point p.

To complete the proof it is enough to show that there is exactly one point in the interior for which all partials vanish. Let us investigate when $\frac{\partial}{\partial b_1}=\ldots=\frac{\partial}{\partial b_k}=0$. If $k=1$ then $\frac{\partial}{\partial b_1}=b_1^{m-1}\left(m\left(b_2-b_1\right)-b_1\right)$ vanishes if and only if $b_1=\alpha_1 b_2$, for $\alpha_1=\frac{m}{m+1}$. Using this equation for $2\leq k\leq d-1$ we have that $\frac{\partial}{\partial b_k}=-b_k^m+mb_k^{m-1}\left(b_{k+1}-b_k\right)+b_{k-1}^m=-b_k^m+mb_k^{m-1}\left(b_{k+1}-b_k\right)+\alpha_{k-1}^m b_k^m=b_k^{m-1}\left(m\left(b_{k+1}-b_k\right)+\alpha_{k-1}^m b_k-b_k\right)$ vanishes if and only if $b_k=\alpha_k b_{k+1}$, for $\alpha_k=\frac{m}{m+1-\alpha_{k-1}^m}$. This establishes a recursive formula for b_i's, and recall that by assumption $b_d=c$. Observe that α_i's are monotonically increasing, $\frac{m}{m+1} = \alpha_1 < \alpha_2 < \ldots < \alpha_{d-1} < 1$, and consequently b_i's are also increasing $0 < b_1 < \ldots < b_{d-1} < b_d = c$. Thus $b_1, \ldots, b_d$ define a unique point in the interior of the domain where all partials vanish. This completes the proof. □

In the remainder of the section we assume that $m \geq 2$ and $d \geq 2$ are constants. Let $i_j = r_j \cdot c$. Observe that then the numbers $r_1, \ldots, r_d$ and $x_1, \ldots, x_d$ defined in the statement of the Lemma 3.4 are some rational numbers that depend only on d and m. Let M be the least common multiple of the denominators of all r_j's. For any c that is a multiple of M the corresponding numbers $i_1, \ldots, i_d$ are natural numbers. We define a Multipartition problem parameterized with M, the fractions $r_1, \ldots, r_d$, and $x_1, \ldots, x_d$.

Multipartition

Instance: A number $c = M \cdot k$, for some natural k, and non-negative rational sizes $s_1, \ldots, s_c$.

Objective: Find a partition $P_1, \ldots, P_d$ of $[c]$ such that $|P_j| = r_j \cdot c$ and $\sum_{k \in P_j} s_k = x_j \cdot \sum s_k$.

Using essentially the same argument as in Lemma 3.2 together with Lemma 3.4 (instead of Lemma 3.1) we can reduce the Multipartition problem to the Conference Call problem restricted to the m and d.

Lemma 3.5. *The Conference Call problem restricted to the $m \geq 2$ and $d \geq 2$ is harder than the Multipartition problem.*

Our goal now is to show that the Multipartition problem is harder than an NP-complete problem. We show this using two reductions.

We first define a variation of the Partition problem and then show that it can be reduced to the Multipartition problem. Let π be a permutation on $[d]$ which sorts the sequence of x_i's in a non-increasing order $x_{\pi(1)} \geq \ldots \geq x_{\pi(d)}$. Let u be the index of the smaller of $r_{\pi(d-1)}$ and $r_{\pi(d)}$ or

$\pi(d)$ if they are equal, and let v be the other index.

Quasipartition2

Instance: A number $n = M(r_u + r_v) \cdot h$, for some natural h, and non-negative rational sizes $s_1, \ldots, s_n$.

Objective: Decide if there a subset P of $[n]$ such that $|P| = M \cdot r_v \cdot h$ and $\sum_{k \in P} s_k = \frac{x_v}{x_u + x_v} \sum s_k$.

Lemma 3.6. *The Multipartition problem is harder than the Quasipartition2 problem.*

Proof. Without loss of generality let us assume that π is the identity permutation (because we can always sort the sequences x_j, r_j, and i_j at the beginning of the proof).

Let $\hat{s}_1, \ldots, \hat{s}_n$ be an instance of the Quasipartition2 problem. We define an instance of the Multipartition problem for $c = \frac{n}{r_u + r_v}$ in a few steps. First we let $s_j = \frac{\hat{s}_j}{\sum \hat{s}_k}(x_{d-1} + x_d)$, for $1 \leq j \leq i_{d-1} + i_d$. Let s be the maximum number which is smaller or equal to any of $s_1, \ldots, s_n$ and also smaller or equal to any difference $x_j - x_{j+1}$ for j for which this difference is non-zero. Next for each x_j, $1 \leq j \leq d-2$, we define i_j additional sizes such that the first "big" is equal to $x_j - \frac{s(i_j - 1)}{2c}$, and the remaining $i_j - 1$ "small" sizes are equal to $\frac{s}{2c}$. Observe that this instance can be constructed in time polynomial to the size of the input sequence.

If the instance of the Quasipartition2 problem has a positive answer then $x_v = \frac{\sum_{k \in P} \hat{s}_k}{\sum \hat{s}_k}(x_u + x_v)$ for some subset of $[n]$ of cardinality Mr_vh. Thus $\sum_{k \in P} s_k = x_v$ and $\sum_{k \in [n] \setminus P} s_k = x_u$, and since $\sum s_k = x_1 + \ldots + x_d = 1$ a multipartition can be found.

Suppose that a multipartition has been found for the instance $s_1, \ldots, s_c$. Observe that for any $1 \leq j \leq d-2$ the first size constructed for x_j is so large that it can only fit into a partition P_k of size x_j. The remaining space in the partition is so small that it can only accommodate exactly $i_j - 1$ "small" sizes $\frac{s}{2c}$. Consequently the cardinality of P_k must be i_j. After possibly swapping partitions P_j with P_k we can assume that $|P_j| = i_j$. We can carry out this process for $j = 1$, $j = 2$, and so on until $j = d - 2$. Consequently exactly two partitions one of size x_d and the other of size x_{d-1} contain respectively i_{d-1} and i_d sizes and these must be all the n sizes $s_1, \ldots, s_n$. As a result there exists a subset P of $[n]$ of cardinality $\max\{i_d, i_{d-1}\}$ such that $\sum_{k \in P} s_k = x_v$, and after multiplying both sides by $\frac{\sum_{k\in[n]} \hat{s}_k}{x_{d-1}+x_d}$ we see that the instance $\hat{s}_1, \ldots, \hat{s}_n$ of the Quasipartitio2 problem has a positive answer.

□

To complete the sequence of reductions and show NP-hardness of the restriction of the Conference Call problem it is enough to show that the Quasipartition2 problem is NP-complete. We will show a reduction from the Partition problem defined earlier (which asks if here is a subset of half of the sizes that sums up to half of the total sum of the sizes).

Lemma 3.7. *The Quasipartition2 problem is NP-complete.*

Proof. Let us first assume that $x_v > x_u$ and let $\hat{S} = \hat{s}_1, \ldots, \hat{s}_g$ be an instance of the Partition problem. We define an instance of the Quasipartition2 problem with $n = M\,(r_u + r_v) \cdot h$ sizes, for $h = 2\lceil \frac{g}{2Mr_u} \rceil$. This n is large enough so that $M \cdot r_v \cdot h - 1 \geq M \cdot r_u \cdot h - 1 \geq \frac{g}{2}$. Let $\bar{u} = Mr_u h - 1 - g/2 \geq 0$, and $\bar{v} = Mr_v h - 1 - g/2 \geq 0$, and $U = \{g+1, \ldots, g+\bar{u}\}$, and $V = \{g+\bar{u}+1, \ldots, g+\bar{u}+\bar{v}\}$. Let $p = \lceil \log_2(\hat{s}_1 + \ldots + \hat{s}_g + 1) \rceil$. We define a list $S = s_1, \ldots, s_n$ as: $s_k = \hat{s}_k + 2^p$, for $1 \leq k \leq g$, and $s_k = 0$, for $k \in U \cup W$. Then we rescale these sizes so that they and two additional sizes $s_{n-1} = \frac{x_v - \frac{1}{3}x_u}{x_u + x_v}$ and $s_n = \frac{2}{3} \cdot \frac{x_u}{x_u + x_v}$ sum to 1 (that is, we take $S = \sum_{k=1}^{g} s_k$ and then let s_k become $\frac{s_k}{S}\,(1 - s_{n-1} - s_n)$, for $1 \leq k \leq g$).

We show that the instance of the Partition problem has a positive answer if and only if the instance of the Quasipartition2 problem has an answer. The forward implication is a simple consequence of the above definitions. Let us investigate the opposite implication.

Suppose that there exists a subset P of cardinality $M \cdot r_v \cdot h$ such that $\sum_{k\in P} s_k = \frac{x_v}{x_u + x_v}$.

Observe that s_{n-1} is strictly larger than the size of the partition $[n] \setminus P$ and so s_{n-1} must belong to P. Moreover the remaining space in the partition P is too small to accommodate s_n and so s_n must be a member of the partition $[n] \setminus P$. But then the remaining space in each partition is equal to exactly $\frac{1}{3} \cdot \frac{x_u}{x_v + x_u}$, and so there exists a subset of $M \cdot r_v \cdot h - 1$ sizes other than the s_{n-1} and the s_n that sum up to half of the total sum of these other sizes. Since the sizes $s_1, \ldots, s_g$ have a summand with factor 2^p, there must exist a subset I of $[g]$ of cardinality $g/2$ that sums up to half of $\sum_{k=1}^{g} \hat{s}_k$, and so the instance of the Partition problem has a positive answer.

If $x_u > x_v$ then the proof is obtained mutatis mutandis. If $x_u = x_v$ then the remaining space in each partition is $1/6$. Thus the result follows. □

Observe that when $M = 3$, $r_u = 1/3$, $r_v = 2/3$, $x_u = x_v = 1/2$ the Quasipartition2 problem becomes the Quasipartition1 problem, and so the latter is NP-complete by the above lemma.

The sequence of reductions shown above leads to the following theorem.

Theorem 3.8. *The restriction of the Conference Call problem to fixed $m \geq 2$ and $d \geq 2$ is NP-hard.*

4 Constant factor approximation

For clarity of presentation, before we show a general approximation result, we discuss a special case of the Conference Call problem with two mobile devices and maximum delay of two.

4.1 Approximation of the special case $m = 2$ and $d = 2$

From Lemma 3.2 we know that this special case is NP-hard. Here we show that there is a $\frac{4}{3}$ approximation solution that can be found in $O(c)$ time and $O(1)$ space. First we show two simple propositions.

Proposition 4.1. *Let $1 \le x \le 2$, and variables a_i and b_i satisfy constraints $a_i, b_i \ge 0$, $a_i + b_i \le 1$, $a_1 + a_2 \ge x - (b_1 + b_2)$, then*

$$(a_1 + b_1)(a_2 + b_2) \ge x - 1 \ .$$

Proof. Notice that by decreasing some a_i the value of the product $(a_1 + b_1)(a_2 + b_2)$ can only be reduced. Thus it is enough to bound the product from below with the additional constraint $a_1 + a_2 = x - (b_1 + b_2)$. But then $(a_1 + b_1)(a_2 + b_2) = (x - (a_2 + b_2))(a_2 + b_2)$, and we can rewrite this expression as $(x - v)v$ for $v = a_2 + b_2$. By the assumption that $a_2 + b_2 \le 1$ we have that $v \le 1$, and by the assumptions that $a_1 + a_2 = x - (b_1 + b_2)$ and that $a_1 + b_1 \le 1$ we have that $v = a_2 + b_2 = x - (a_1 + b_1) \ge x - 1$. Note that for the domain $x - 1 \le v \le 1$, the expression $(x - v)v$ treated as a function of v is minimized at the boundary of the domain when $v = x - 1$, or when $v = 1$. This yields a lower bound of $x - 1$. □

Proposition 4.2. *Let $0 < s \le c$ and $1 \le x \le 2$, then $c - s(x-1) \le \frac{4}{3}\left(c - s\left(\frac{x}{2}\right)^2\right)$.*

Proof. Since $\frac{\partial^2}{\partial x^2}\left(c - s(x-1) - \frac{4}{3}\left(c - s\left(\frac{x}{2}\right)^2\right)\right) = \frac{2}{3}s > 0$, the function is a strictly convex function of x, and so it is bounded from above by the greater of the values achieved for $x = 1$ or for $x = 2$. For $x = 2$ its value is $c - s - \frac{4}{3}(c - s) < 0$, and for $x = 1$ its value is $c - \frac{4}{3}\left(c - \frac{1}{4}s\right) \le c - \frac{4}{3} \cdot \frac{3}{4}c = 0$. Thus the function is bounded from above by 0, and the result follows. □

Suppose that $\langle S_1, S_2 \rangle$ is a strategy that minimizes expected paging among all strategies of length 2. We do not know how to find this strategy, but despite this let us assume that we know $|S_1| = s_1$. Let us pick s_1 cells T_1 which maximize the expected number of devices located in the cells i.e., for any cell j in T_1 and any cell j' not in T_1 we have $p_{1,j} + p_{2,j} \ge p_{1,j'} + p_{2,j'}$. Then the following lemma shows that the strategy $\langle T_1, T_2 \rangle$ has desired approximation, where $T_2 = [c] \setminus T_1$.

Lemma 4.3. *$EP_T / EP_S \le 4/3$.*

Proof. Recall that by the selection of the strategy $\langle T_1, T_2 \rangle$ the set T_1 contains cells j that maximize the sum $p_{1,j} + p_{2,j}$. We divide the cells in $S_1 \cup T_1$ into three sets $B = S_1 \cap T_1$, $A = T_1 \setminus B$, and $C = S_1 \setminus B$. For each mobile device i we define a_i to be the probability that the device is in the cells A i.e., $a_i = \sum_{j \in A} p_{i,j}$. Similarly we define $b_i = \sum_{j \in B} p_{i,j}$ and $c_i = \sum_{j \in C} p_{i,j}$.

By the theorem of the arithmetic and geometric means [14] we have $\prod_{i=1}^{2} \sum_{j \in S_1} p_{i,j} \le \left(\frac{x}{2}\right)^2$, where $x = (b_1 + c_1) + (b_2 + c_2)$, $0 \le x \le 2$. Recall that for each device i we have $p_{i,1} + \ldots + p_{i,c} = 1$, consequently $a_i + b_i \le 1$. By the choice of T_1 we have that $a_1 + a_2 \ge c_1 + c_2 = x - (b_1 + b_2)$.

We divide the analysis of the approximation ratio into two cases. If $x < 1$ then we bound the probability that all devices are found by the strategy $\langle T_1, T_2 \rangle$ in the first round by 0. (Recall that s_1 is the number of cells paged in the first round). Thus

$$\frac{EP_T}{EP_S} \le \frac{c}{c - (c - s_1) \prod_{i=1}^{2} \sum_{j \in S_1} p_{i,j}} \le \frac{c}{c - (c - s_1)\frac{1}{4}} \le \frac{c}{c - c\frac{1}{4}} = \frac{4}{3} \; .$$

For the second case assume that $1 \le x \le 2$. By Proposition 4.1 we can bound the ratio as

$$\frac{EP_T}{EP_S} \le \frac{c - (c - s_1)(x - 1)}{c - (c - s_1)\left(\frac{x}{2}\right)^2} \; .$$

It remains to show that the enumerator is bounded from above by $\frac{4}{3}$ times the denominator. But this follows from Proposition 4.2, which completes the proof. □

Note that we do not need to know the size of S_1 because we can afford to evaluate expected paging of a strategy $\langle T_1, T_2 \rangle$ constructed for $s_1 = 1$, then evaluate it for $s_1 = 2$, and so on until $s_1 = c - 1$, and pick the strategy that minimizes expected paging. Thus we can find a strategy that is a $\frac{4}{3}$ approximation without the knowledge of the size of the set of cells paged in the first round by a two-round strategy that minimizes expected paging.

4.2 General case

This section is devoted to generalizing the ideas given in the previous section. For this purpose we employ techniques from multidimensional convex optimization and dynamic programming. Specifically we show an $\frac{e}{e-1}$-approximation algorithm for the Conference Call problem. Our algorithm uses a very simple heuristic. Intuitively a good paging strategy first pages cells that have large chances of finding all mobile devices in the cells. To realize this intuition we sequence cells in a non-increasing order of the sum of probabilities of finding mobile devices in the cells. In particular the first cell in the sequence is a cell j that maximizes $p_{1,j} + \ldots + p_{m,j}$ and the last cell in the sequence is a cell j that minimizes $p_{1,j} + \ldots + p_{m,j}$. Then we find a certain strategy that pages, in each round, consecutive cells from the sequence and show that this strategy is a desired approximation. The proof of the correctness of the approximation algorithm has two steps. We consider a family of all strategies of length d that page cells according to the sequence. First we show that there exists a strategy T in this family that has the value of expected paging by at most $\frac{e}{e-1}$ factor greater than the minimal value of expected paging for any strategy (not necessarily from the family). However the strategy T needs to probe, in each round, the same number of cells as a strategy S that minimizes expected paging. Since we do not know the sizes of groups for the strategy S we cannot efficiently find T! In the second step we show that with the help of dynamic programming we can efficiently find a strategy G in the family that minimizes expected paging across all strategies in the family. Thus the strategy G has expected paging at most equal to expected paging of the strategy T, which completes the proof. In the following two sections we present the details of this argument.

In the remainder of the section we assume that all cells are ordered so that for all $1 \leq j < j' \leq c$ $p_{1,j} + \ldots + p_{m,j} \geq p_{1,j'} + \ldots + p_{m,j'}$. We consider a family $\mathcal{F}$ of strategies such that a strategy

$\langle S_1, \ldots, S_d \rangle$ from the family has the property that any group S_j, $j \geq 2$, contains cells greater than all preceding groups in the strategy i.e., for all $1 \leq i < j \leq d$, for all $i' \in S_i$ and $j' \in S_j$, we have $i' < j'$.

4.2.1 Existence of approximate solution

In this section we demonstrate that the family $\mathcal{F}$ contains a strategy T that has the expected number of cells paged until all mobile devices are found larger by at most the factor of $\frac{e}{e-1}$ than the expected paging of a strategy of length d that minimizes the expectation. The key idea behind the proof is to observe that if an optimal strategy yields small chances of finding all mobile devices until a round then the strategy T is trivially good enough, while when these chances are high then the strategy T must also have high chance of finding all mobile devices.

For the analysis of performance of our heuristic we need two technical inequalities, which we show next. These inequalities generalize Proposition 4.1 and Proposition 4.2 given in the previous section. We prove the more general facts using techniques from multidimensional convex optimization.

Lemma 4.4. *Let $m \geq 2$, $m - 1 \leq x \leq m$, and variables a_i and b_i satisfy constraints $a_i, b_i \geq 0$, $a_i + b_i \leq 1$, $\sum_{i=1}^{m} a_i \geq x - \sum_{i=1}^{m} b_i$ then*

$$\prod_{i=1}^{m} (a_i + b_i) \geq x - m + 1 \ .$$

Proof. Notice that the value of the product $\prod_{i=1}^{m} (a_i + b_i)$ can only be reduced by decreasing some a_i. Thus it is enough to bound the product from below with additional constraint $\sum_{i=1}^{m} a_i = x - \sum_{i=1}^{m} b_i$.

The proof is by induction on m. The base case for $m = 2$ was shown in Proposition 4.1. For the inductive step take $\sum_{i=1}^{m+1} (a_i + b_i) = x$, for $m \leq x \leq m + 1$. Observe that $m - 1 \leq$

$x-(a_{m+1}+b_{m+1}) \leq m$. Thus we can use the inductive hypothesis to bound the product by

$$\prod_{i=1}^{m+1}(a_i+b_i) \geq (x-(a_{m+1}+b_{m+1})-m+1)\,(a_{m+1}+b_{m+1}) \ .$$

Since $x-m \leq a_{m+1}+b_{m+1} \leq 1$ the result follows. □

Lemma 4.5. *Let $x_1,\ldots,x_k$ be variables satisfying $m-1 \leq x_i \leq m$, for some $m \geq 2$. Let $s_2,\ldots,s_d$, $k \leq d-1$, be positive and satisfy $s_2+\ldots+s_d \leq c$, for some $c \leq 0$ then*

$$c-\sum_{r=1}^{k} s_{r+1}\,(x_r-m+1) \leq \frac{e}{e-1}\left(c-\sum_{r=1}^{k} s_{r+1}\left(\frac{x_r}{m}\right)^m - \frac{s_{k+2}+\ldots+s_d}{e}\right) \ .$$

Proof. We consider a function

$$f(x_1,\ldots,x_k) = c-\sum_{r=1}^{k} s_{r+1}\,(x_r-m+1) + \frac{e}{e-1}\left(-c+\sum_{r=1}^{k} s_{r+1}\left(\frac{x_r}{m}\right)^m + \frac{s_{k+2}+\ldots+s_d}{e}\right)$$

defined over domain $H=[m-1,m]^k$ (i.e., k-dimensional cube), and show that its maximum is at most 0, which will complete the proof.

For this purpose let us extend the domain of f to $D=(m-1-\epsilon,m+\epsilon)^k$, for some $0<\epsilon<1$. Observe that D is an open and convex subset of $\mathbb{R}^k$, and that f has continuous second partial derivatives in D. Also for every point $x \in D$, the Hessian matrix $H(x)$ of f is diagonal, and that the entries on the diagonal are strictly positive $\frac{\partial^2 f}{\partial x_r^2} = \frac{e s_{r+1} m(m-1)}{(e-1)m^m} x_r^{m-2} > 0$. Thus $H(x)$ is positive definite, and so by Theorem 2.2 f is strictly convex. Since H is a closed subset of D, Lemma 2.3 ensures that the values of f on H are bounded from above by the values achieved at the boundary of H.

After these preliminary observations let us prove the statement of the lemma by induction on k. For the base case assume that $k=1$. There are two boundary cases. When $x_1=m$ then

$$f(x_1) = c-s_2+\frac{e}{e-1}\left(-c+s_2+\frac{s_3+\ldots+s_d}{e}\right) = \frac{s_2+\ldots+s_d-c}{e-1} \leq 0 \ .$$

When $x_1 = m-1$ then

$$f(x_1) = c + \frac{e}{e-1}\left(-c + s_2\left(\frac{m-1}{m}\right)^m + \frac{s_3+\ldots+s_d}{e}\right) \leq \frac{s_2+\ldots+s_d-c}{e-1} \leq 0\ .$$

Thus f is bounded from above by 0.

For the inductive step we consider the values of f at the boundary of H. Due to symmetricity of the function and constraint conditions we can focus on two cases $x_k = m$ and $x_k = m-1$. When $x_k = m$ then

$$\begin{aligned} f &= c - \sum_{r=1}^{k-1} s_{r+1}\left(x_r - m + 1\right) + \frac{e}{e-1}\left(-c + \sum_{r=1}^{k-1} s_{r+1}\left(\frac{x_r}{m}\right)^m + s_{k+1}\left(1 - \frac{e-1}{e}\right) + \frac{s_{k+2}+\ldots+s_d}{e}\right) \\ &= c - \sum_{r=1}^{k-1} s_{r+1}\left(x_r - m + 1\right) + \frac{e}{e-1}\left(-c + \sum_{r=1}^{k-1} s_{r+1}\left(\frac{x_r}{m}\right)^m + \frac{s_{k+1}+\ldots+s_d}{e}\right)\ , \end{aligned}$$

and we can use the inductive hypothesis to bound f from above. For the second case consider $x_k = m-1$. Then

$$f \leq c - \sum_{r=1}^{k-1} s_{r+1}\left(x_r - m + 1\right) + \frac{e}{e-1}\left(-c + \sum_{r=1}^{k-1} s_{r+1}\left(\frac{x_r}{m}\right)^m + \frac{s_{k+1}+\ldots+s_d}{e}\right)\ ,$$

and we can use the inductive hypothesis again to bound f from above. □

Suppose that we have guessed the sizes of groups paged in each round by a strategy that minimizes expected number of cells paged until all mobile devices are found, and let $s_1, \ldots, s_d$ be these sizes (i.e., the strategy pages s_r cells in round r, $s_1 + \ldots + s_d = c$). In the next lemma we demonstrate that when we use our heuristic with the same sizes of groups paged in corresponding rounds then the expected paging may be greater by at most factor $\frac{e}{e-1}$.

Lemma 4.6. *Let $s_1, \ldots, s_d$ be positive integers such that $s_1 + \ldots + s_d = c$. Let $\langle S_1, \ldots, S_d \rangle$ be any strategy for which $|S_r| = s_r$, and let $\langle T_1, \ldots, T_d \rangle$ be the strategy in $\mathcal{F}$ that has groups of the same cardinality $|T_r| = s_r$. Then the ratio of expected number of cells paged until all mobile devices are found is*

$$\frac{EP_T}{EP_S} \leq \frac{e}{e-1}\ .$$

Proof. When $m = 1$ then the ratio $\frac{EP_T}{EP_S} \le 1$, which was studied by other researchers [13, 18, 19]. Our proof focuses on case when $m \ge 2$.

Recall that by the construction the set $Z_r = T_1 \cup \ldots \cup T_r$ contains cells j that maximize the sum $p_{1,j} + \ldots + p_{m,j}$. Let us define $U_r = S_1 \cup \ldots \cup S_r$. For any r we divide the cells in $U_r \cup Z_r$ into three sets $B_r = Z_r \cap U_r$, $A_r = Z_r \setminus B_r$, and $C_r = U_r \setminus B_r$. For each mobile device i we define $a_{i,r}$ to be the probability that the device i is in the cells A_r i.e., $a_{i,r} = \sum_{j \in A_r} p_{i,j}$. Similarly we define $b_{i,r} = \sum_{j \in B_r} p_{i,j}$ and $c_{i,r} = \sum_{j \in C_r} p_{i,j}$.

By the theorem of the arithmetic and geometric means [14] we have $\prod_{i=1}^{m} \sum_{j \in U_r} p_{i,j} \le \left(\frac{x_r}{m}\right)^m$, where $x_r = \sum_{i=1}^{m} (b_{i,r} + c_{i,r})$, $0 \le x_r \le m$. Recall that for each mobile device i we have $p_{i,1} + \ldots + p_{i,c} = 1$, consequently $a_{i,r} + b_{i,r} \le 1$. By the choice of Z_r we have that $\sum_{i=1}^{m} a_{i,r} \ge \sum_{i=1}^{m} c_{i,r} = x_r - \sum_{i=1}^{m} b_{i,r}$.

If $x_r < m - 1$ then we can bound the product $|T_{r+1}| \prod_{i=1}^{m} \sum_{j \in Z_r} p_{i,j}$ from below by 0, and the product $|S_{r+1}| \prod_{i=1}^{m} \sum_{j \in U_r} p_{i,j}$ from above by $|S_{r+1}| \left(\frac{m-1}{m}\right)^m \le |S_{r+1}| e^{-1}$.

If $m - 1 \le x_r \le m$ then by Lemma 4.4 we can bound the product $|T_{r+1}| \prod_{i=1}^{m} \sum_{j \in Z_r} p_{i,j}$ from below by $|T_{r+1}| (x_r - m + 1)$, and the product $|S_{r+1}| \prod_{i=1}^{m} \sum_{j \in U_r} p_{i,j}$ from above by $|S_{r+1}| \left(\frac{x_r}{m}\right)^m$.

Recall that by Lemma 2.1

$$\frac{EP_T}{EP_S} = \frac{c - |T_2| \prod_{i=1}^{m} \sum_{j \in Z_1} p_{i,j} - \ldots - |T_d| \prod_{i=1}^{m} \sum_{j \in Z_{d-1}} p_{i,j}}{c - |S_2| \prod_{i=1}^{m} \sum_{j \in U_1} p_{i,j} - \ldots - |S_d| \prod_{i=1}^{m} \sum_{j \in U_{d-1}} p_{i,j}} .$$

Without loss of generality let us assume that among $x_1, \ldots, x_{d-1}$ only the first k of them satisfy $m - 1 \le x_r \le m$, and the remaining satisfy $0 \le x_r < m - 1$. Thus we can bound the ratio from above by

$$\frac{EP_T}{EP_S} \le \frac{c - s_2 (x_1 - m + 1) - \ldots - s_{k+1} (x_k - m + 1)}{c - s_2 \left(\frac{x_1}{m}\right)^m - \ldots - s_{k+1} \left(\frac{x_k}{m}\right)^m - (s_{k+2} + \ldots + s_d) e^{-1}} .$$

It remains to show that the enumerator is bounded from above by $\frac{e}{e-1}$ times the denominator. If $k = 0$ then the bound is trivial, so assume that $k \ge 1$. But then by Lemma 4.5 the enumerator is never greater than $\frac{e}{e-1}$ times the denominator. Thus the expected paging of the strategy $\langle T_1, \ldots, T_d \rangle$ is at most $\frac{e}{e-1}$ times the expected paging of the strategy $\langle S_1, \ldots, S_d \rangle$. $\square$

This lemma immediately tells us that for any fixed number of rounds d we can find an approximate paging strategy in polynomial time because there are $O(c^{d-1})$ strategies that satisfy $s_1 + \ldots + s_d = c$ (because any selection of values for some $d-1$ variables determines the value of the remaining one variable). However this method is not satisfactory when d and c grow, and so we seek a more scalable solution. We achieve this by using dynamic programming.

4.2.2 Finding approximate solution using dynamic programming

Now our goal is to find a strategy in $\mathcal{F}$ that minimizes expected paging among all strategies in $\mathcal{F}$. For this purpose we develop a dynamic programming algorithm that generalizes an approach given by [13].

Take any $1 \leq \ell \leq k \leq c$ and consider a class of strategies of length ℓ that may page the last k cells and only these cells during the ℓ rounds. Let $\langle S_1, \ldots, S_\ell \rangle$ be a strategy from this class (so that $S_1 \cup \ldots \cup S_\ell = \{c-k+1, \ldots, c-1, c\}$). Let P be a random variable equal to the number of cells paged by this strategy given that at least one mobile device is located among the last k cells (then P is at least 1). In the class we can find a strategy that minimizes the expected value of P and denote this value by $E(\ell, k)$.

Observe that the value of $E(d, c)$ is exactly the minimal expected paging across all strategies in $\mathcal{F}$. We need to show that we can efficiently find a strategy that achieves $E(d, c)$, and as the first step we prove a recursive formula for finding the value of $E(\ell, k)$.

Lemma 4.7. *The value of $E(\ell, k)$, $1 \leq \ell \leq k \leq c$, is equal to:*

$$E(1, k) = k \ ,$$

when $1 \le k \le c$, *and*

$$E(\ell, k) =$$

$$\min_{1 \le x \le k-\ell+1} \left\{ x + \frac{1 - \prod_{i=1}^{m} \sum_{j=1}^{c-k+x} p_{i,j}}{1 - \prod_{i=1}^{m} \sum_{j=1}^{c-k} p_{i,j}} E(\ell - 1, k - x) \right\} ,$$

when $2 \le \ell \le k \le c$.

Proof. We need to show that in each of the two equations above the expression on the right side of the equal sign is equal to the expression on the left side of the sign. The first equation is trivial because any strategy of length 1 that pages exactly k cells during 1 round given that there is at least one mobile device among these k cells has expected number of cell paged equal to k.

In order to show the second equation we prove two inequalities. First we show that the expression on the right side of the equal sign is never greater than $E(\ell, k)$. Let $\langle S_1, \ldots, S_\ell \rangle$ be a strategy that may page the last k cells and only these cells during ℓ rounds and that minimizes the expected value of the number of cells P paged during the ℓ rounds given that at least one mobile device is located in one of the last k cells. Observe that $1 \le |S_1| \le k - \ell + 1$ because none of the groups can be empty. Let A be the event that at least one mobile device is located among the last k cells, and let B be the event that at least one mobile device is located among the last $k - |S_1|$ cells. Observe that the expected value of P is equal to

$$E(\ell, k) = \mathsf{Exp}[P] = |S_1| + \mathsf{Pr}[\ B \mid A\ ] \cdot E(\ell - 1, k - |S_1|) \ .$$

But

$$\mathsf{Pr}[\ B \mid A\ ] = \left(1 - \prod_{i=1}^{m} \sum_{j=1}^{c-k+|S_1|} p_{i,j}\right) / \left(1 - \prod_{i=1}^{m} \sum_{j=1}^{c-k} p_{i,j}\right) ,$$

and so the expression on the right side of the second equation is at most $E(\ell, k)$.

To show that the expression on the right side of the second equation is never smaller than $E(\ell, k)$ observe that for any x, $1 \le x \le k - \ell + 1$, the value of the expression is equal to the

01 approximation(in: $c, m, d, p_{i,j}$,

02 $\quad 1 \le i \le c$, $1 \le j \le m$;

03 $\quad$ out: g_r, $1 \le r \le d$)

04 array

05 $\quad X[1,\ldots,d;1,\ldots,c]$, $F[1,\ldots,c]$

06 $\quad E[1,\ldots,d;1,\ldots,c]$, $S[1,\ldots,m]$

07 for $i = 1$ to m

08 $\quad S[i] = 0$

09 for $j = 1$ to c

10 $\quad$ for $i = 1$ to m

11 $\quad\quad S[i] = S[i] + p_{i,j}$

12 $\quad F[j] = 1$

13 $\quad$ for $i = 1$ to m

14 $\quad\quad F[j] = F[j] \cdot S[i]$

15 for $k = 1$ to c

16 $\quad E[1,k] = k$

17 $\quad X[1,k] = k$

18 for $\ell = 2$ to d

19 $\quad$ for $k = \ell$ to c

20 $\quad\quad E[\ell,k] = \infty$

21 $\quad\quad$ for $x = 1$ to $k - \ell + 1$

22 $\quad\quad\quad v = x + \frac{1-F[c-k+x]}{1-F[c-k]} \cdot E[\ell-1, k-x]$

23 $\quad\quad\quad$ if $v < E[\ell,k]$ then

24 $\quad\quad\quad\quad E[\ell,k] = v$

25 $\quad\quad\quad\quad X[\ell,k] = x$

26 $w = c$

27 for $\ell = d$ downto 1

28 $\quad g_{d-\ell+1} = X[\ell,w]$

29 $\quad w = w - X[\ell,w]$

Figure 1: Algorithm for finding the sizes $g_1,\ldots,g_d$ of groups for a strategy that achieves $\frac{e}{e-1}$ approximatation factor.

expected number of cells paged by a strategy of length ℓ that may page the last k cells and only these cells, x of which in the first round, given that at least one mobile device is among the last k cells. Hence the value of the expression can never be smaller than the value $E(\ell,k)$ that is the minimum of the expectation. This completes the proof of the lemma. □

During the calculation of $E(d,c)$ we can find the sizes of groups for a strategy $\langle G_1,\ldots,G_d\rangle$ that has expected paging equal to $E(d,c)$ (see Figure 1 for a pseudocode of an algorithm. Lines 07 through 14 calculate the probabilities that all devices are found by round r, for $r = 1, \ldots, c$. Lines 15 through 25 evaluate the recursive formula given in Lemma 4.7. Lines 26 through 29 find the sizes of groups for the approximation strategy that were calculated during the evaluation of the recursive formula). This leads to the main theorem of the paper.

Theorem 4.8. *For any instance of the Conference Call problem the strategy $\langle G_1, \ldots, G_d \rangle$ has expected paging*

$$EP_G \leq \frac{e}{e-1} EP_{MIN} \ ,$$

and it can be found in $O\left(c\left(m + dc\right)\right)$ time and $O(m + dc)$ space.

Proof. Take any instance of the Conference Call problem. By Lemma 4.6 the family $\mathcal{F}$ contains a strategy $\langle T_1, \ldots, T_d \rangle$ that has expected paging at most $\frac{e}{e-1}$ times the minimal expected paging of any strategy of length d. By Lemma 4.7 the dynamic programming finds a strategy $\langle G_1, \ldots, G_d \rangle$ that minimizes expected paging across all strategies in the family. Thus the expected paging of the strategy $\langle G_1, \ldots, G_d \rangle$ is at most $\frac{e}{e-1}$ times the expected paging of any strategy of length at most d. □

We remark that the dynamic programming approach developed in this section allows us to find a strategy that minimizes expected paging across a family of strategies that page cells in any predefined sequence.

4.3 Lower bound on performance

Finally we give a lower bound on the performance ratio of our heuristic. Consider an instance of the Conference Call problem with $m = 2$, $c = 8$, and $d = 2$. Let $p_{1,1} = \frac{2}{7}$, $p_{2,1} = p_{1,7} = p_{1,8} = 0$, and the remaining probabilities are set to $\frac{1}{7}$. By simple case analysis we can show that the best strategy pages cells 2 through 6 in the first round and achieves $\frac{317}{49}$ expected paging, while the heuristic chooses to page cells 1 through 5 in the first round and achieves $\frac{320}{49}$ expected paging. This establishes a lower bound of $\frac{320}{317}$ on the performance ratio of our heuristic.

5 Future work and discussion

In this section we list some open problems and report some of our results on our work in progress and then mention some related work.

Our approximation solution has a small ≈ 1.58 approximation factor and cannot be better than $\frac{320}{317}$. Is there a better approximation algorithm or even an approximation scheme? So far we know an approximation scheme for a subclass of the problem [7]. Here we assume that the set of probabilities $\{p_{i,j} : 1 \leq i \leq m, 1 \leq j \leq c\}$ can be covered by a constant number of real intervals of constant length. This allows us to search the space of solutions exhaustively in polynomial time.

In this paper we consider only oblivious strategies i.e., where the set of cells to be probed in each round is fixed in advance. It is interesting to also consider adaptive strategies which determine, in each round, the set of cells to page depending on the devices found in earlier rounds. Our NP-hardness result applies to adaptive strategies as well since for $d = 2$ any adaptive strategy is oblivious. One can easily extend the heuristic presented in Section 4 to form an adaptive strategy where, in each round, we calculate conditional probabilities and based on their values we determine the group of cells to page in the next round using the algorithm presented in Figure 1. The analysis of the performance ratio of the resulting algorithm stands as an open problem.

There are other interesting types of searches to consider. A dual problem to the Conference Call problem is the *Yellow Pages* problem in which we are searching for one out of m possible devices. We showed [7] an m-approximation algorithm based on a heuristic that is different from the one considered in this paper. We also know that the heuristic considered in this paper does not offer constant factor approximation. A problem that generalizes the two problems is the *Signature* problem where we are looking for any k devices out of the m devices. The conference call problem is the case where $k = m$ and the yellow pages problem is the case where $k = 1$. Solutions to the signature problem can be applied to the task of finding k managers out of m managers to sign a document.

Another interesting direction is to extend the model. For example due to bandwidth limitations in real systems it may be reasonable to assume that at most a fixed number of b cells can be paged at any unit of time. Here we observe that our approximation result generalizes to yield results in this model: we can use Lemma 4.6 to show the existence of an approximate strategy, and in Lemma 4.7 we can limit the range for x accordingly and consider only $E(\ell, k)$ when $k \leq b\ell$ to find an appropriate strategy. Another possible extension of the model is to assume that when there is a device at a cell and we page the cell we do not always find out if the device is there (similar assumptions have been considered [21]), and that the chances of finding out decrease with the increase of the number of devices in the cell. This models collision of response signals to the paging signal emitted from the base station.

5.1 Related work

Burkard et al. [8] review a combinatorial optimization problem related to the Conference Call problem. The problem is called the Quadratic Assignment Problem [15] and it is formulated as follows. Given two symmetric matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ of size c by c with non-negative entries find a permutation π that maximizes $\sum_{i,j} a_{i,j} b_{\pi(i),\pi(j)}$. It can be shown [7] that one can use a solution to the Quadratic Assignment Problem to solve the Conference Call problem for two mobile devices. If d is constant then the reduction is polynomial time.

There are several results from Search Theory [21] that can be used in location management.

Search Theory deals with finding a single object located among a set of cells as given by a probability distribution. Searching consists of a sequence of *lookups* of the cells. There is a cost associated with looking up a cell. It is assumed that with some probability a lookup may not find an object in a cell even though the object is located there. The goal is to decide which cells and when to look up to maximize chances of finding the object under constraints on the total cost of search. Awduche et al. [3] consider a model where fixed groups of cells are paged and a probability is given that paging a cell does not detect a device even though it is located at the cell. Authors show how to apply results from Search Theory to minimize the expected number of cells paged.

Another related search problem is the Combinatorial Group Testing problem [9]. Using the terminology of our paper, the model of CGT assumes that a paging of a subset of cells (called items in CGT) returns whether a mobile device (called defective in CGT) exists among the cells paged. Thus the search may need to continue recursively by repaging smaller and smaller sets in order to locate the single cell where the mobile device exists (is located). In the model that we study in this paper we assume that a paging of any subset of cells returns all the devices that are located in one of the cells that have been paged and for each of the devices that have been found the cell where the device is located. Hence no recursive repaging is needed.

Acknowledgements. The work of the second author would not be possible without generous support and encouragement from his manager Michael Merritt while an intern at AT&T Shannon Lab, whom he would like to thank. The second author would also like to thank his advisor Alex Shvartsman for very valuable discussions. The authors would like to thank David Johnson, Jeff Lagarias, and S. Muthukrishnan for discussion, and the anonymous PODC reviewers for comments that improved the presentation of the paper.

References

[1] Abutaleb, A., Li, V.O.K.: Location update optimization in personal communication systems. Wireless Networks, Vol. 3(3) (1997) 205–216

[2] Akyildiz, I.F., McNair, J., Ho, J.S.M., Uzunalioglu, H., Wang, W.: Mobility Management in Next Generation Wireless Systems. IEEE Proceedings Journal, Vol. 87(8) (1999) 1347-1385

[3] Awduche, D.O., Gaylord, A., Ganz, A.: On Resource Discovery in Distributed Systems with Mobile Hosts. MOBICOM'96 (1996)

[4] Bar-Noy, A.: Quantum Directories – a Search Engine for Mobile Users. A manuscript.

[5] Bar-Noy, A., Kessler, I., Sidi, M.: Mobile Users: To Update or not to Update? Wireless Networks journal, Vol. 1 (1995) 175186

[6] Bar-Noy, A., Kessler, I.: Tracking Mobile Users in Wireless Networks. IEEE Transactions on Information Theory, Vol. 39 (1993) 1877–1886

[7] Bar-Noy, A., Malewicz, G.: Establishing Wireless Conference Calls. Manuscript, AT&T proprietary (2001)

[8] Burkard, R.E., Cela, E., Pardalos, P.M., Pitsoulis L.S.: The Quadratic Assignment Problem. In Handbook of Combinatorial Optimization (D.-Z. Du and P.M. Pardalos, eds.), Vol. 2, Kluwer Academic Publishers (1998) 241–337

[9] Du, D.-Z., Hwang, F.K.: Combinatorial Group Testing and Its Applications (2nd Edition). World Scientific, Series on Applied Mathematics Vol. 12 (1999)

[10] EIA/TIA: Cellular radio-telecommunications intersystem operations. EIA/TIA Technical Report IS-41 Revision C (1995)

[11] ETSI/TC: Mobile application part (MAP) specification, version 4.8.0. Technical Report, Recommendation GSM 09.02 (1994)

[12] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)

[13] Goodman, D., Krishnan, P., Sugla., B.: Minimizing Queuing Delays and Number of Messages in Mobile Phone Location. ACM/Baltzer J. on Mobile Networks and Applications, Vol. 1(1) (1996) 39–48

[14] Hardy, G.H., Littlewood, J.E., Pólya, G.: Inequalities. Cambridge University Press (1964)

[15] Koopmans, T.C., Beckermann, M.J.: Assignment problems and the location of economic activities. Econometrica, Vol. 25 (1957) 53–76

[16] Kostrikin, A.I., Manin, Yu.I.: Linear Algebra and Geometry. Gordon & Breach Science Pub (1989)

[17] Liu, T., Bahl, P., Chlamtac, I.: Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks. IEEE Journal on Special Areas in Communications, Special Issue on Wireless Access Broadband Networks, Vol. 16(6) (1998) 922–936

[18] Madhavapeddy, S., Basu, K., Roberts, A.: Adaptive Paging Algorithms for Cellular Systems. Wireless Information Networks: Architecture, Resource Management and Mobile Data (1996) 83–101

[19] Rose, C., Yates, R.: Minimizing The Average Cost of Paging Under Delay Constraints. ACM Journal of Wireless Networks, Vol. 1(2) (1995) 211–219

[20] Stoer, J., Witzgall, Ch.: Convexity and Optimization in Finite Dimensions. Springer-Verlag (1970)

[21] Stone, L.D.: Theory of Optimal Search. Academic Press, 1975