Algorithmica (2004) 40: 293–304
DOI: 10.1007/s00453-004-1114-1

Algorithmica

Admission Control in Networks with Advance Reservations

Liane Lewin-Eytan,[1] Joseph (Seffi) Naor,[2] and Ariel Orda[1]

Abstract. The provisioning of quality-of-service for real-time network applications may require the network to reserve resources. A natural way to do this is to allow advance reservations of network resources prior to the time they are needed. We consider several two-dimensional admission control problems in simple topologies such as a line and a tree. The input is a set of connection requests, each specifying its spatial characteristics, that is, its source and destination; its temporal characteristics, that is, its start time and duration time; and, potentially, also a bandwidth requirement. In addition, each request is associated with a profit gained by accommodating it. We address the related admission control problem, where the goal is to maximize the total profit gained by the accommodated requests. We provide approximation algorithms for several problem variations. Our results imply a $4c$-approximation algorithm for finding a maximum weight independent set of axis-parallel rectangles in the plane, where c is the size of a maximum set of overlapping rectangles.

Key Words. Line network, Approximation algorithms, Independent set, Local ratio, Axis parallel rectangles, Advance reservations.

1. Introduction

1.1. *Problem Statement and Motivation.* As network capabilities increase, their usage is also expanding. At the same time, the wide range of requirements of the many applications using them calls for new mechanisms to control the allocation of network resources. However, while much attention has been devoted to resource reservation and allocation, the same does not apply to the timing of such requests. In particular, the prevailing assumption has been that requests are "immediate", i.e., made at the same time the network resources are needed. This is a useful base model, but it ignores the possibility, present in many other resource allocation situations, that resources might be requested in *advance* of when they are needed. This can be a useful service, not only for applications, which can then be sure that the resources they need will be available, but also for the network, as it enables better planning and more flexible management of resources. Accordingly, advance reservation of network resources has been the subject of several recent studies and proposals, e.g., [14], [17], [18], and [21]. It has also been recognized that some of the related algorithmic problems are hard [14].

We investigate some fundamental admission control problems in networks with advance reservations. We concentrate on networks having special topologies such as lines and trees. Yet, even for these topologies, the problems we consider remain NP-hard.

[1] Department of Electrical Engineering, Technion, Haifa 32000, Israel. liane@tx.technion.ac.il. ariel@ee.technion.ac.il.
[2] Department of Computer Science, Technion, Haifa 32000, Israel. naor@cs.technion.ac.il.

Received March 2003; revised October 2003. Communicated by S. Leonardi.
Online publication August 4, 2004.

These problems are in essence two-dimensional: we are presented with commodities (connection requests), each having a spatial dimension determined by its route in the specific topology, and a temporal dimension determined by its future duration. Each request specifies its source and destination, start time and duration time, and potentially also a bandwidth requirement. In addition, each request is associated with a profit gained by accommodating it. In the line and tree topologies, each request from a source s to a destination d has only one possible path, thus, the routing issue is non-existent. The admission control problem that we address is which of the requests should be accommodated. The goal is to maximize the total profit gained by the accommodated requests. The optimal solution consists of a feasible set of requests with maximum total profit. We consider several variants of the problems described above, all of which are NP-hard, and provide approximation algorithms for all of them.

1.2. *Previous Work and Our Contribution*. Some of the earliest work on advance reservation in communication networks was done in the context of video-conferencing and satellite systems. Early video-conferencing systems involved high bandwidth signals between (fixed) video-conferencing sites (studios), and advance reservations were needed to ensure that adequate bandwidth was available. Similarly, early satellite systems offered the option to rent the use of transponders for specific amounts of time, which also required support for advance reservation. In those early systems the bulk of the work (e.g., [16] and [20]) focused on traffic modeling and related call admission procedures, to size such facilities properly. Some more recent studies (e.g., [17] and [21]) have extended these early works from the circuit-switched environment they assumed to that of modern integrated packet switching networks. Most other studies dealing with advance reservation in networks have focused on extensions to signalling protocols, or formulated frameworks (including signalling and resource management capabilities) to support advance reservations, e.g., [15] and [18]. The routing perspective of networks with advance reservations has been investigated in [14]. That study considered possible extensions to path selection algorithms in order to make them advance-reservation aware; as connections were assumed to be handled one at a time, the admission control problem was trivial. Competitive analysis of on-line admission control in general networks was studied in [4].

In this study we focus on advance reservations of *multiple connections* in specific network topologies, namely, lines and trees. The case of advance reservations in a line topology can be modeled as a set of axis-parallel rectangles in the plane: the x-axis represents the links of the network, and the y-axis represents the time line which is assumed to be slotted. Each rectangle corresponds to a request: its projection on the x-axis represents its path from the source to the destination, and its projection on the y-axis represents its time interval.

For a set R of n axis-parallel rectangles in the plane, the associated *intersection graph* is the undirected graph with vertex set equal to R and an edge between two vertices if the corresponding rectangles intersect. Assuming each rectangle (corresponding to some vertex in the intersection graph) has a profit associated with it, the goal is to find the *maximum weight independent set* (MWIS) in the intersection graph. That is, to find a set of non-overlapping rectangles with maximum total profit. This problem has already been considered in the context of label placement in digital cartography. The task is to place labels on a map, where the labels can be modeled as rectangles. They are assigned profit values that represent the importance of including them in the map. Often, a label can be placed in more than one position on the map. The goal is thus to compute an MWIS of labels.

An approximation algorithm for the MWIS problem on axis-parallel rectangles achieving a factor of $O(\log n)$ was given in [1]. This factor was improved to $\log n/\alpha$ for any constant α in [9]. For the case where all rectangles have the same height (in our model, the same duration of time), a 2-approximation algorithm that incurs $O(n \log n)$ time is given in [1]. Extending this result, using dynamic programming, a PTAS, i.e., a $(1+1/k)$-approximation algorithm whose running time is $O(n \log n + n^{2k-1})$, for any $k \geq 1$, is obtained [1].

For the two-dimensional problem on a tree topology, we get an approximation algorithm by investigating the one-dimensional case, where only the spatial dimension is considered. Several variants of this problem have been studied in the literature, e.g., [10]–[13].

1.2.1. *Our Results*. For line topologies, i.e., for the MWIS problem in the intersection graph of axis-parallel rectangles, we obtain a $4c$-approximation algorithm, where c denotes the maximum number of rectangles that can simultaneously cover a point in the plane. This improves on the approximation factor in [1] and [9] for the case where c is small (c is $o(\log n)$). Our technique for deriving the $4c$-approximate solution implies the following result. Suppose that two rectangles are defined to be intersecting if and only if one of them covers a corner of the other. A 4-approximation for the MWIS problem is obtained in this case by using the local structure of the linear programming relaxation of the problem. It remains an intriguing open problem whether a constant approximation factor can be found for the MWIS problem in an intersection graph of arbitrary axis-parallel rectangles in the plane.

For a tree topology, we consider the case where each request has a bandwidth demand that is defined to be the *width* of the request. We first present a 5-approximation algorithm for the MWIS problem in the case of one-dimensional requests in a tree (that is, their durations are ignored). Then we provide an $O(\log n)$-approximation for the two-dimensional case. (These factors also hold for the case of a line topology.)

1.3. *Model*. The time domain over which reservations are made is composed of *time slots* $\{0, 1, 2, \ldots\}$ of equal size. The duration of each reservation is an integer number of slots. In both topologies (lines and trees) the available bandwidth of each link $l \in E$ is fixed over time (before any requests are being accommodated); we normalize it to unit size for convenience. We are presented with a set R of n *connection requests* (also termed *commodities*), each specifying its source and destination nodes, and a specific amount of bandwidth B from some time slot t_1 up to some time slot t_2, $t_2 > t_1$. Two cases are considered: the case where all demands B are equal to 1 (that is, only a single request can be routed through link l during time slot t), and the case where the demands are arbitrary numbers bounded by 1. Each request I has a profit $p(I)$ gained by routing it. The goal is to select a *feasible* set of requests with maximum total profit. A set is *feasible* if for all time slots t and for all links l, the total bandwidth of requests whose time interval contains t and whose route contains l does not exceed 1.

2. Preliminaries: The Local Ratio Technique. We use the *local ratio* technique [6] extensively, hence we briefly present some related preliminaries.

Let $\mathbf{p} \in \mathbb{R}^n$ be a profit (or penalty) vector, and let F be a set of feasibility constraints on vectors $\mathbf{x} \in \mathbb{R}^n$. A vector $\mathbf{x} \in \mathbb{R}^n$ is a *feasible solution* to a given problem $(F, \mathbf{p})$ if it satisfies all of the constraints in F. The *value* of a feasible solution $\mathbf{x}$ is the inner product $\mathbf{p} \cdot \mathbf{x}$. A feasible solution is *optimal* for a maximization (or minimization) problem if its value is maximal (or minimal) among all feasible solutions. A feasible solution $\mathbf{x}$ is an *r-approximate* solution, or simply an *r-approximation*, if $\mathbf{p} \cdot \mathbf{x} \geq r \cdot \mathbf{p} \cdot \mathbf{x}^*$ (or $\mathbf{p} \cdot \mathbf{x} \leq r \cdot \mathbf{p} \cdot \mathbf{x}^*$ when $\mathbf{p}$ is a penalty vector), where $\mathbf{x}^*$ is an optimal solution. An algorithm is said to have a *performance guarantee* of r if it always computes r-approximate solutions.

The local ratio technique was introduced in [7] and later extended in [5], [6], and [8]. It relies on the following theorem.

THEOREM 1 (Local Ratio). *Let F be a set of constraints, and let $\mathbf{p}$, $\mathbf{p_1}$, $\mathbf{p_2}$ be profit (or penalty) vectors where $\mathbf{p} = \mathbf{p_1} + \mathbf{p_2}$. If $\mathbf{x}$ is an r-approximate solution with respect to $(F, \mathbf{p_1})$ and with respect to $(F, \mathbf{p_2})$, then $\mathbf{x}$ is an r-approximate solution with respect to $(F, \mathbf{p})$.*

An algorithm that uses the local ratio technique typically proceeds as follows. Initially, the solution is empty. The idea is to find a decomposition of $\mathbf{p}$ into $\mathbf{p_1}$ and $\mathbf{p_2}$ such that $\mathbf{p_1}$ is an "easy" weight function in some respect, e.g., any solution that is maximal with respect to containment would be a good approximation to the optimal solution of $(F, \mathbf{p_1})$. The local ratio algorithm recursively continues on the instance $(F, \mathbf{p_2})$. We inductively assume that the recursively returned solution for the instance $(F, \mathbf{p_2})$ is a good approximation and need to prove that it is also a good approximation for $(F, \mathbf{p})$. This requires proving that the recursively returned solution for the instance $(F, \mathbf{p_2})$ is also a good approximation for the instance $(F, \mathbf{p_1})$. This step is usually the core of the proofs on approximation factors.

3. Line Topology. The case of advance reservations in a line topology can be modeled as a set of axis-parallel rectangles in the plane: the x-axis represents the network links, and the y-axis represents the time line. Each rectangle corresponds to a request: its projection on the x-axis represents its path from the source to the destination, and its projection on the y-axis represents its time interval. We consider the case where all bandwidth requirements are equal to the capacity of the links.

For a set R of n rectangles in the plane, the associated *intersection graph* $G = (R, E)$ is the undirected graph with a vertex set equal to R and an edge between two vertices if and only if the corresponding rectangles intersect. We assume that each rectangle (corresponding to some vertex in the intersection graph) has a *profit* (or *weight*) associated with it. The goal is to find an MWIS in G, i.e., a set of non-overlapping rectangles with maximum total profit.

The problem of finding a maximum independent set in the intersection graph of unit squares is NP-complete [2]. Since unit squares are a special case of rectangles, the intractability of this problem implies the intractability of the general case.

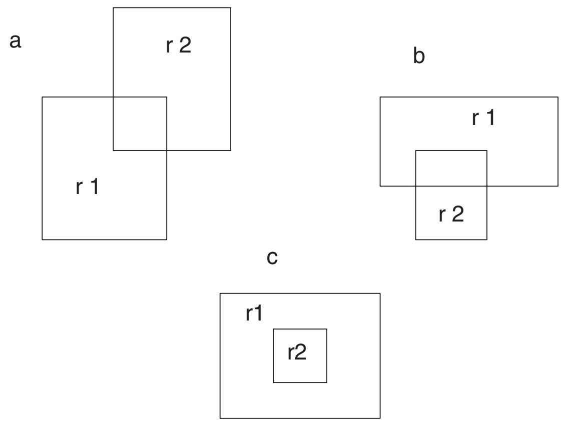

Fig. 1. In (a) both rectangles are vertex-intruding into each other, while in (b) and (c) r_2 is vertex-intruding into r_1.

A maximal (with respect to containment) clique Q in G corresponds to a point in the plane a such that the vertices in Q correspond to the rectangles covering a. We assume that the maximum clique in G is of size c, i.e., no point is covered by more than c rectangles.

A rectangle r_1 is said to be *vertex-intruding* into another rectangle r_2 if r_2 contains at least one of the corners of r_1. Two rectangles are *vertex-incident* if at least one of them is vertex-intruding into the other. Two identical rectangles are also said to be vertex-intruding into each other. Figure 1 contains several examples of vertex-incident rectangles. A set of rectangles is said to be *vertex-incident-free* if no two of them are vertex-incident. For a rectangle r, we denote by $N^i[r]$ the set of rectangles that are vertex-incident to r. (Note that $r \in N^i[r]$.)

We present a $4c$-approximation algorithm for the MWIS problem. A high-level description of the algorithm is as follows:

1. Compute a set S of rectangles that are vertex-incident-free, such that the weight of S is at least one-quarter of the weight of an MWIS in G.
2. Find an independent set $I \subseteq S$ of rectangles, such that its weight is at least $1/c$ of the weight of S.
3. The output is the independent set I.

Clearly, the independent set I is a $4c$-approximate solution. We now elaborate on the steps of the algorithm.

Step 1. We formulate the following linear program (L) for the MWIS in G. We define an indicator variable $x(v)$ for each rectangle $v \in R$. If $x(v) = 1$, then rectangle v belongs to the independent set. The linear relaxation of the indicator variables assigns fractions to the rectangles (requests) under the constraint that for each clique Q, the sum of the fractions assigned to all rectangles in Q does not exceed 1. Let $\mathbf{x}$ denote the vector of indicator variables.

(L) $$\text{Maximize} \sum_{v \in R} w(v) \cdot x(v)$$

subject to:

(1) $$\text{for each clique } Q, \quad \sum_{v \in Q} x(v) \leq 1,$$

(2) $$\text{for all } v \in R, \quad x(v) \geq 0.$$

Note that the number of cliques in G is $O(n^2)$. It is easy to see that an independent set in G provides a feasible integral solution to the linear program. Thus, the value of an optimal (fractional) solution to the linear program is an upper bound on the value of an optimal integral solution.

We compute an optimal solution to (L). We now show how to round the solution obtained to get the set S. The core of our rounding algorithm is the following lemma.

LEMMA 2. *Let* $\mathbf{x}$ *be a feasible solution to* (L). *Then there exists a rectangle* $v \in R$ *satisfying*

$$\sum_{u \in N^i[v]} x(u) \leq 4.$$

PROOF. For two vertex-incident rectangles u and v, define $y(u, v) = x(v) \cdot x(u)$. Also, define $y(u, u) = x(u)^2$. For a point a in the plane, let $C(a)$ denote the set of rectangles that contain a. For a rectangle r, let $S(r)$ denote the set of four corners of r. We prove the lemma using a *weighted average* argument, where the weights are the values $y(u, v)$ for all pairs of vertex-incident rectangles, u and v. Specifically, we claim that

(3) $$\sum_{v \in R} \sum_{u \in N^i[v]} y(u, v) \leq \sum_{v \in R} \sum_{a \in S(v)} \sum_{u \in C(a)} y(u, v).$$

We establish (3) by considering the different cases of vertex-incidence between two rectangles u and v (see Figure 1). If u and v are vertex-incident, then they contribute together $2y(u, v)$ to the left-hand side of (3). The contribution to the right-hand side of (3) is as follows:

1. Rectangle u is vertex-intruding into v and vice versa (see Figure 1(a)). In this case, u and v contribute $2y(u, v)$ to the right-hand side of (3), since u (v) contains a corner of v (u).
2. Rectangle v is vertex-intruding into u, but u is not vertex-intruding into v (see Figure 1(b)). In this case, v contributes $2y(u, v)$ to the right-hand side of (3), since v has two corners intruding into u.
3. Rectangle v is either contained inside rectangle u, or $v = u$ (see Figure 1(c)). In this case, v contributes $4y(u, v)$ to the right-hand side of (3), since u contains all four corners of v.

Hence, there exists a rectangle v satisfying

$$\sum_{u \in N^i[v]} y(u, v) \leq \sum_{a \in S(v)} \sum_{u \in C(a)} y(u, v).$$

If we factor out $x(v)$ from both sides, we obtain

$$\sum_{u \in N^i[v]} x(u) \leq \sum_{a \in S(v)} \sum_{u \in C(a)} x(u).$$

From constraint (1) in (L), it follows that, for each rectangle v and all $a \in S(v)$, $\sum_{u \in C(a)} x(u) \leq 1$. Therefore,

$$\sum_{u \in N^i[v]} x(u) \leq 4,$$

thus completing the proof. □

We now use a fractional version of the local ratio technique developed in [8]. The proof of the next lemma is immediate.

LEMMA 3 (Fractional Local Ratio). *Let* $\mathbf{x}$ *be a feasible solution to* (L). *Let* $\mathbf{w_1}$ *and* $\mathbf{w_2}$ *be a decomposition of the weight vector* $\mathbf{w}$ *such that* $\mathbf{w} = \mathbf{w_1} + \mathbf{w_2}$. *Suppose that* $\mathbf{z}$ *is a feasible integral solution vector to* (L) *satisfying* $\mathbf{w_1} \cdot \mathbf{z} \geq r(\mathbf{w_1} \cdot \mathbf{x})$ *and* $\mathbf{w_2} \cdot \mathbf{z} \geq r(\mathbf{w_2} \cdot \mathbf{x})$. *Then*

$$\mathbf{w} \cdot \mathbf{z} \geq r(\mathbf{w} \cdot \mathbf{x}).$$

The rounding algorithm will apply a local ratio decomposition of the weight vector $\mathbf{w}$ with respect to an optimal solution $\mathbf{x}$ to the linear program (L). The algorithm for computing S proceeds as follows:

1. Delete all rectangles with non-positive weight. If no rectangles remain, return the empty set.
2. Let $v' \in R$ be a rectangle satisfying $\sum_{u \in N^i[v']} x(u) \leq 4$. Decompose $\mathbf{w}$ by $\mathbf{w} = \mathbf{w_1} + \mathbf{w_2}$ as follows:

$$w_1(u) = \begin{cases} w(v') & \text{if} \quad u \in N^i[v'], \\ 0 & \text{otherwise.} \end{cases}$$

   (In the decomposition the component $\mathbf{w_2}$ may be non-positive.)
3. Solve the problem recursively using $\mathbf{w_2}$ as the weight vector. Let S' be the returned independent set.
4. If v' is not vertex-incident to any rectangle in S', return $S = S' \cup \{v'\}$; otherwise, return $S = S'$.

Clearly, the set S is vertex-incident-free. We now analyze the quality of the solution produced by the algorithm.

THEOREM 4. *Let* $\mathbf{x}$ *be an optimal solution to the linear program* (L). *Then it holds for the set* S *computed by the algorithm that* $w(S) \geq \frac{1}{4} \cdot \mathbf{w} \cdot \mathbf{x}$.

PROOF. The proof is by induction on the number of recursive calls. At the basis of the recursion, the returned set satisfies the theorem, since no rectangles remain. Clearly, the first step in which rectangles of non-positive weight are deleted cannot decrease the above right-hand side. We now prove the inductive step. Let $\mathbf{z}$ and $\mathbf{z}'$ be the indicator vectors of the sets S and S', respectively. Assume that $\mathbf{w_2} \cdot \mathbf{z}' \geq 1/4 \cdot \mathbf{w_2} \cdot \mathbf{x}$. Since $w_2(v') = 0$, it also holds that $\mathbf{w_2} \cdot \mathbf{z} \geq 1/4 \cdot \mathbf{w_2} \cdot \mathbf{x}$. From step 4 of the algorithm it follows that at least one vertex from $N^i[v']$ belongs to S. Hence, $\mathbf{w_1} \cdot \mathbf{z} \geq 1/4 \cdot \mathbf{w_1} \cdot \mathbf{x}$. Thus, by Lemma 3, it follows that $\mathbf{w} \cdot \mathbf{z} \geq 1/4 \cdot \mathbf{w} \cdot \mathbf{x}$, i.e., $w(S) \geq \frac{1}{4} \cdot \mathbf{w} \cdot \mathbf{x}$. □

Step 2. The input to this step is a set of rectangles S that are vertex-incident-free. We show that the intersection graph of such a family of rectangles is a perfect graph, i.e., a graph whose chromatic number, as well as those of all of its subgraphs, are equal to the size of the maximum clique.

Let the maximum clique size in S be c'. Clearly, $c' \leq c$.

THEOREM 5. *There exists a legal coloring of the rectangles in S that uses c' colors precisely.*

PROOF. The proof is based on ideas from [3]. Rectangles that are not vertex-incident can only intersect in the pattern described in Figure 2. Let us look at the cliques of size c' in S and choose, in each clique, the tallest and narrowest rectangle. We observe that these rectangles are disjoint, otherwise one of them is vertex-intruding into the other. We color the rectangles chosen by the same color. Since the maximum clique among the remaining rectangles in S is of size $c' - 1$, we can continue this process recursively, and get a c'-coloring of S. □

The proof of the above theorem is constructive, namely, we color the rectangles of S by c' colors and choose I to be the color class of maximum weight, thus,

$$w(I) \geq \frac{w(S)}{c'} \geq \frac{w(S)}{c}.$$

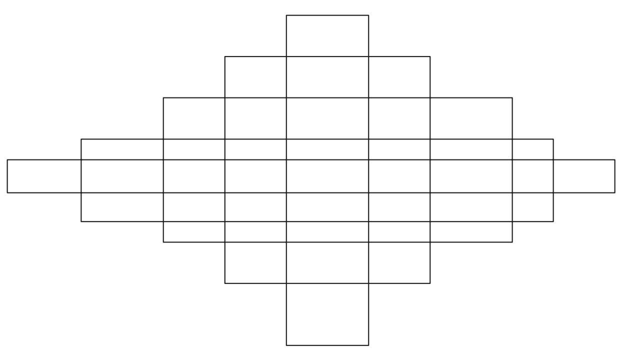

Fig. 2. Intersection of rectangles that are not vertex-incident.

4. Tree Topology. In the previous section we have assumed that each connection requested the full link capacity. We now consider the case where each request r_i has a bandwidth demand $0 < w_i \leq 1$, which is defined to be the *width* of the request. We present a $(5 \log n)$-approximation algorithm. First, we present a 5-approximation algorithm for the MWIS problem in the case of one-dimensional requests in a tree (that is, their durations are ignored). Then we provide a $(5 \log n)$-approximation for the two-dimensional case by extending the algorithm presented in [1], which is a simple divide-and-conquer algorithm for computing a maximum independent (non-overlapping) set of n axis-parallel rectangles in the plane. The one-dimensional case in tree topologies has received much attention, e.g., [10]–[13]. However, to the best of our knowledge, there are no known approximation factors prior to our work for the one-dimensional problem we consider here, i.e., where requests have widths.

We proceed to present a 5-approximation algorithm for the one-dimensional MWIS problem in a tree. We divide our instances (requests) into two sets: a set consisting of all *narrow* instances, i.e., that have width of at most $\frac{1}{2}$, and a set consisting of all *wide* instances, i.e., that have width greater than $\frac{1}{2}$. We solve our problem separately for the two sets, and return the solution with greater profit.

The problem where all instances are wide reduces to the problem of finding an MWIS of paths in a tree, since no pair of intersecting paths can be in the solution simultaneously. This problem can be solved optimally (see [19]).

We proceed to describe a 4-approximation algorithm for the case where all instances are narrow, as follows. Define the *least common ancestor* of a path to be the vertex (in the path) that is closest to the root, where the distance between two vertices is defined to be the number of edges in the path connecting them.

1. Delete all instances with non-positive profit.
2. If no instances remain, return the empty set. Otherwise, proceed to the next step.
3. Let $\tilde{I}$ be a path whose least common ancestor has maximum distance from the root.
4. Decompose the profit vector p by $p = p_1 + p_2$.
5. Solve the problem recursively using p_2 as the profit function. Let S' be the returned set.
6. If $S' \cup \{\tilde{I}\}$ is a feasible set, return $S = S' \cup \{\tilde{I}\}$. Otherwise, return $S = S'$.

Define $\mathcal{I}(I)$ to be the set of instances intersecting I ($\mathcal{I}(I)$ includes I). The profit decomposition is the following:

$$p_1(I) = p(\tilde{I}) \cdot \begin{cases} 1, & I = \tilde{I}, \\ \alpha \cdot w(I), & I \in \mathcal{I}(\tilde{I}), \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

THEOREM 6. *The algorithm presented above, with* $\alpha = 1/(1 - w(\tilde{I}))$, *yields a* 4-*approximate solution*.

PROOF. A feasible set is called I-*maximal* if it either contains instance I, or otherwise adding I to it will render it infeasible. Define b_{opt} to be an upper bound on the optimum p_1-profit and define $b_{\max}$ to be a lower bound on the p_1-profit of every $\tilde{I}$-maximal set, both are normalized by $p(\tilde{I})$. We consider an optimal solution. By the definition of p_1, only instances in $\mathcal{I}(\tilde{I})$ contribute to its p_1-profit. We denote the least common ancestor vertex of $\tilde{I}$ by v. Path $\tilde{I}$ can have at most two edges l_1 and l_2 that are adjacent to v. Since $\tilde{I}$ is a path whose least common ancestor has maximum distance from the root, it can intersect with at most two disjoint groups of paths: one group contains the paths intersecting l_1, and the other group contains the paths intersecting l_2. For each edge $l \in \{l_1, l_2\}$, the total width of instances in $\mathcal{I}(\tilde{I})$ that use l in their path is at most 1. Therefore, in case $\tilde{I}$ does not belong to the optimal solution, the contribution of the other instances in $\mathcal{I}(\tilde{I})$ is at most $2\alpha \cdot p(\tilde{I})$. Otherwise, if $\tilde{I}$ is in the optimal solution, the contribution of the other instances in $\mathcal{I}(\tilde{I})$ is at most $2\alpha(1 - w(\tilde{I})) \cdot p(\tilde{I})$. Thus, we get $b_{\text{opt}} = \max\{2\alpha, 1 + 2\alpha(1 - w(\tilde{I}))\}$.

Turning to $\tilde{I}$-maximal solutions, either such a solution contains $\tilde{I}$ or else it contains a set $\mathcal{X} \neq \emptyset$ of instances intersecting $\tilde{I}$ that prevent $\tilde{I}$ from being added to the solution. The total width of instances in $\mathcal{X}$ is at least $1 - w(\tilde{I})$, for otherwise $\tilde{I}$ can be added to the solution. We thus have $b_{\max} = \min\{1, \alpha \cdot (1 - w(\tilde{I}))\}$.

The approximation factor of the algorithm in the case where all paths are narrow is at least

$$\frac{\min\{1, \alpha \cdot (1 - w(\tilde{I}))\}}{\max\{2\alpha, 1 + 2\alpha(1 - w(\tilde{I}))\}}. \tag{5}$$

For $\alpha = 1/(1 - w(\tilde{I}))$ we get an approximation factor of $\frac{1}{4}$. □

COROLLARY 7. *By choosing the solution with greater profit out of the two sets* (*wide and narrow*), *we get an approximation factor of* $\frac{1}{5}$ *for the general one-dimensional case*.

We now present a $(5 \log n)$-approximation for the two-dimensional case. Let R be the set of n requests. We sort the requests by their start and end time-coordinates. Let t_{med} be the median time-coordinate. That is, the number of requests whose time-coordinates are below or above t_{med} is not more than $n/2$. We partition the requests of R into three groups, R_1, R_2, and R_{12}, as follows. R_1 and R_2 contain the requests whose time-coordinates are respectively below and above t_{med}. R_{12} contains the requests with time interval containing time slot t_{med}, and thus defines a one-dimensional problem: as all these requests intersect in the time line, their time intervals can be ignored. Now, we compute the approximate MWIS M_{12} of R_{12} (as explained before). We recursively compute M_1 and M_2, the approximate MWIS in R_1 and R_2, respectively: if $p(M_{12}) \geq p(M_1) + p(M_2)$, then we return M_{12}, otherwise we return $M_1 \cup M_2$ ($p(M)$ denotes the sum of the profits of the requests in M).

THEOREM 8. *The algorithm computes a* $(5 \log n)$*-approximate solution*. *Generally*, *for any network topology*, *a* ρ*-approximation algorithm for the one-dimensional problem implies a* $(\rho \log n)$*-approximation for the two-dimensional problem*.

PROOF. The proof is by induction. For $n \leq 2$, we can optimally compute a maximum weight set, and the claim follows. Suppose it holds for all $m < n$. Let M^* be the optimal solution for R. Similarly, let M_1^*, M_2^*, and M_{12}^* be the optimal solutions of R_1, R_2, and R_{12}, respectively. As stated in Corollary 7 , there is a 5-approximation algorithm for the one-dimensional problem, thus we have $p(M_{12}) \geq p(M_{12}^*)/5 \geq p(M^* \cap R_{12})/5$. By the inductive hypothesis,

$$p(M_1) \geq \frac{p(M_1^*)}{5\log(n/2)} \geq \frac{p(M^* \cap R_1)}{5(\log n - 1)}, \quad \text{and, similarly,} \quad p(M_2) \geq \frac{p(M^* \cap R_2)}{5(\log n - 1)}.$$

Thus,

$$\begin{aligned} p(M) &= \max\{p(M_{12}), p(M_1) + p(M_2)\} \\ &\geq \max\left\{\tfrac{1}{5}p(M^* \cap R_{12}), \frac{p(M^* \cap R_1) + p(M^* \cap R_2)}{5(\log n - 1)}\right\} \\ &\geq \max\left\{\tfrac{1}{5}p(M^* \cap R_{12}), \frac{p(M^*) - p(M^* \cap R_{12})}{5(\log n - 1)}\right\}. \end{aligned}$$

If $p(M^* \cap R_{12}) \geq p(M^*)/\log n$, the inductive step is established, otherwise

$$\frac{p(M^*) - p(M^* \cap R_{12})}{5(\log n - 1)} \geq \frac{p(M^*) - p(M^*)/\log n}{5(\log n - 1)} = \frac{p(M^*)}{5\log n}$$

and the inductive step is established as well. □

References

[1] P.K. Agarwal, M. van Kreveld, and S. Suri, Label placement by maximum independent set in rectangles, *Computational Geometry*: *Theory and Applications*, **11**(3–4): (1998), 209–218.

[2] T. Asano, Difficulty of the maximum independent set problem on intersection graphs of geometric objects, *Graph Theory*, *Combinatorics*, *and Applications*, *Vol*. 1, *Kalamazoo*, *MI*, 1988, Wiley-Interscience, New York, 1991, pp. 9–18.

[3] E. Asplund and B. Grunbaum, On a coloring problem, *Mathematica Scandinavica* **8** (1960), 181–188.

[4] B. Awerbuch, Y. Azar, and S. Plotkin, Throughput-competitive online routing, *Proceedings of the* 34*th Annual Symposium on Foundations of Computer Science*, 1993, pp. 32–40.

[5] V. Bafna, P. Berman, and T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problem, *SIAM Journal on Discrete Mathematics*, **12**, (1999), 289–297.

[6] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber, A unified approach to approximating resource allocation and scheduling, *Journal of the ACM*, **48** (2001), 1069–1090.

[7] R. Bar-Yehuda and S. Even, A local-ratio theorem for approximating the weighted vertex cover problem, *Annals of Discrete Mathematics*, **25**, (1985) 27–46.

[8] R. Bar-Yehuda, M. Halldorsson, J. Naor, H. Shachnai, and I. Shapira, Scheduling split intervals, *Proceedings of the* 13*th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2002, pp. 732–741.

[9] P. Berman, B. DasGupta, S. Muthukrishnan, and S. Ramaswami, Improved approximation algorithms for rectangle tiling and packing, *Proceedings of the* 12*th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001, pp. 427–436.

[10] C. Chekuri, M. Mydlarz, and F.B. Shepherd, Multicommodity demand flow in a tree, *Proceedings of the* 30*th International Colloquium on Automata*, *Languages*, *and Programming*, 2003, pp. 410–425.

[11] T. Erlebach, Call admission control for advance reservation requests with alternatives, *Proceedings of the* 3*rd Workshop on Approximation and Randomization Algorithms in Communication Networks*, Proceedings in Informatics, Carleton Scientific, Waterloo, Ontario, 2002, pp. 51–64.

[12] T. Erlebach and K. Jansen, Off-line and on-line call-scheduling in stars and trees, *Proceedings of the* 23*rd International Workshop on Graph-Theoretic Concepts in Computer Science* (*WG* ’97), LNCS 1335, Springer-Verlag, Berlin, 1997, pp. 199–213.

[13] T. Erlebach and K. Jansen, Maximizing the number of connections in optical tree networks, *Proceedings of the Ninth Annual International Symposium on Algorithms and Computation* (*ISAAC* '98), LNCS 1533, Springer-Verlag, Berlin, 1998, pp. 179–188.

[14] R. Guerin and A. Orda, Networks with advance reservations: the routing perspective, *Proceedings of INFOCOM* '2000, Tel-Aviv, 2000, Vol. 1, pp. 118–127.

[15] W. Reinhardt, Advance resource reservation and its impact on reservation protocols, Proceedings of Broadband Islands '95, Dublin, September 1995. A technical report can be found at citeseer.ist.psu.edu/reinhardt95advance.html.

[16] J. Roberts and K. Liao, Traffic models for telecommunication services with advance capacity reservation, *Computer Networks and ISDN Systems*, **10**(3–4) (1985), 221–229. (Special issue: selected papers from the 11th International Teletraffic Congress.)

[17] O. Schelén and S. Pink, Sharing resources through advance reservations agents, *Proceedings of IFIP 5th International Workshop on Quality of Service*, New York, 1997, pp. 265–276.

[18] A. Schill, F. Breiter, and S. Kühn, Design and evaluation of an advance resource reservation protocol on top of RSVP, *Proceedings of the 4th IFIP International Conference on Broadband Communications*, Stuttgart, 1998, pp. 23–40.

[19] R.E. Tarjan, Decomposition by clique separators, *Discrete Mathematics*, **55** (1985), 221–231.

[20] J. Virtamo, A model of reservation systems, *IEEE Transactions on Communications*, **40**(1) (1992) 109–118.

[21] D. Wischik and A. Greenberg. Admission control for booking ahead shared resources, *Proceedings of INFOCOM* '98, San Francisco, CA, 1998, pp. 873–882.