# Routing and Admission Control in Networks with Advance Reservations

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

[1] Electrical Engineering Dept., Technion, Haifa 32000, Israel.
E-mail: liane@tx.technion.ac.il, ariel@ee.technion.ac.il,
[2] Computer Science Dept., Technion, Haifa 32000, Israel
E-mail: naor@cs.technion.ac.il

**Abstract.** The provisioning of quality-of-service (QoS) 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, a ring 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 has a profit gained by acommodating 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 requests.

## 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 as when 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, 18, 19, 22]. 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, rings and trees. Yet, even for these topologies, the problems we consider remain NP-hard. 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 has 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, while in an undirected ring topology, each request has two possible routes, namely clockwise and counter-clockwise. Thus, in all cases, the routing issue is either simple or nonexistent. We address the following admission control problem: which of the requests will 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 we 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., [17, 21]) focused on traffic modeling and related call admission procedures, to properly size such facilities. Some more recent studies (e.g., [18, 22]) have extended these early works from the circuit-switched environment they assumed to that of modern integrated packet switching networks. Most other works 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, [16, 19]. 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, rings 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. 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 the maximum weight independent set of label objects.

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

**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 cover simultaneoulsy a point in the plane. This improves on the approximation factor of [1, 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 an undirected ring topology, we can generalize the same approximation factors obtained for a line topology. We also consider the case where the ratio between the duration of different requests does not differ by more than $k$ a parameter. We present a $\log k$-approximation algorithm for this case.

For a tree topology, we consider the case where each request has a bandwidth demand which 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 cases of line and ring topologies.)

### 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 all three topologies (lines, rings 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 (=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 $t_2 > t_1$. We consider two cases: the case where all demands $B$ are equal to one (that is, only a single request can be routed through link $l$ during time slot $t$), and the case where the demands are positive arbitrary numbers bounded by one. 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 shall 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 $\mathbf{p} \cdot \mathbf{x} \geq$ (or $\leq$) $r \cdot \mathbf{p} \cdot \mathbf{x}^*$, 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.

Our algorithms use the local ratio technique. This technique was first developed by Bar-Yehuda and Even [7] and later extended by [5, 6, 8]. The local ratio theorem is as follows.

**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 which is maximal with respect to containment would be a good approximation to the optimal solution of $(F, \mathbf{p_1})$. The local ratio algorithm continues recursively on the instance $(F, \mathbf{p_2})$. We assume inductively that the solution returned recursively 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 solution returned recursively for the instance $(F, \mathbf{p_2})$ is also a good approximation for the instance $(F, \mathbf{p_1})$. This step is usually the "heart" of the proof of the approximation factor.

## 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 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 a *maximum weight independent set* (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.

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 the rectangles are not pairwise 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. The algorithm is structured as follows.

1. Compute a set $S$ of rectangles that are vertex-incident-free, such that the weight of $S$ is at least $1/4$ of the weight of a 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.

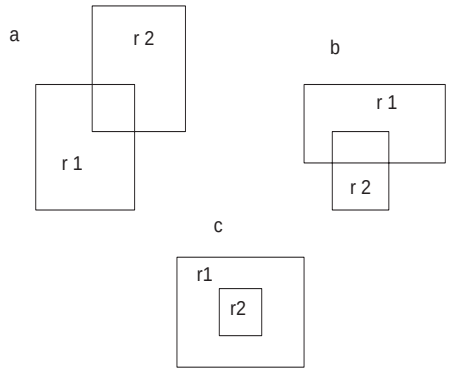



**Fig. 1.** In (a) both rectangles are vertex-intruding into each other, while in (b) and (c) $r_2$ is vertex-intruding into $r_1$.

*Step (1).* We formulate a linear program 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) with 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.

$$\text{(L)} \quad \text{maximize} \qquad \sum_{v \in R} w(v) \cdot x(v)$$

*subject to:*

$$\text{For each clique } Q\text{:} \sum_{v \in Q} x(v) \leq 1 \tag{1}$$

$$\text{For all } v \in R\text{:} \quad x(v) \geq 0. \tag{2}$$

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 1.** *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 vertices $u$ 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 $v$. We claim that

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

We shall prove (3) by considering the different cases of vertex-incidence between two rectangles $u$ and $v$ (see Fig. 1). If $u$ and $v$ are vertex-incident, then they contribute together $2y(u,v)$ to the LHS of Equation (3). The contribution to the RHS of Equation (3) is as follows.

1. Rectangle $u$ is vertex-intruding into $v$ and vice versa (see Fig. 1(a)). In this case, $u$ and $v$ contribute $2y(u,v)$ to the RHS of Equation (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 Fig. 1(b)). In this case, $v$ contributes $2y(u,v)$ to the RHS of Equation (3), since $v$ has two corners intruding into $u$.
3. Rectangle $v$ is either contained inside rectangle $u$, or $v = u$ (see Fig. 1(c)). In this case, $v$ contributes $4y(u,v)$ to the RHS of Equation (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 $v$, for all $a \in S(v)$, $\sum_{u \in C(a)} x(u) \leq 1$. Therefore,

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

Completing the proof. ♠

We now use a fractional version of the Local Ratio technique developed by [8]. The proof of the next lemma is immediate.

**Lemma 2 (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 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:

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

   (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 independent set returned.
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 2.** *Let* $\mathbf{x}$ *be an optimal solution to 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 set returned satisfies the theorem, since no rectangles remain. Clearly, the first step in which rectangles of non-positive weight are deleted cannot decrease the above RHS. 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 2, 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. Perfect graphs are graphs for which the chromatic number is equal to the maximum clique size and this property holds for all subgraphs.

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

**Theorem 3.** *There exists a legal coloring of the rectangles in* $S$ *that uses precisely* $c'$ *colors.*

*Proof.* We use here 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,

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

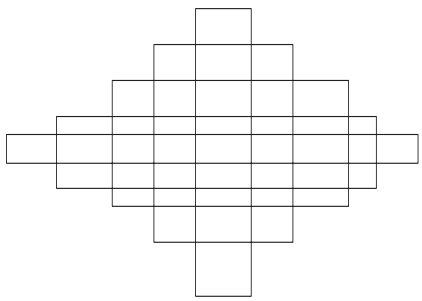


**Fig. 2.** Intersection of rectangles that are not vertex-incident

### 3.1 Ring Topology

We consider an undirected ring, that is, each request has two possible routes from its source to its destination, namely clockwise and counter-clockwise. Each request has a profit associated with it, and the goal is to find a MWIS of requests, that is, a set of non-overlapping requests with maximum total profit. We have to choose a unique path for each request such that the profit of the solution is maximized. We consider the case where all bandwidth requirements are equal to the capacity of the links.

We first remark that the approximation factors obtained for the case of a line topology can be generalized for this case too. We now consider the case where the ratio between the duration of different requests is not more than $k$, a parameter. We sketch how to obtain a $\log k$-approximation algorithm in this case. We start with a line. Partition the set of rectangles into groups such that the heights are within a factor of two in each group. There are at most $\log k$ groups. For each group, a constant-factor approximate solution can be computed using ideas from [1] (and also [12]). The output is the best of the $\log k$ solutions, yielding an approximation factor of $\log k$. This approach can be generalized to rings by cutting one link and then reducing the problem to the line case.

## 4 Tree Topology

In previous sections we 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. 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 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. We remark that these results also hold for the cases of line and ring topologies. The one dimensional case in tree topologies has received much attention, e.g., [10, 11]. However, to the best of our knowledge, there are no known approximation factors for the one-dimensional problem we are considering here (where requests have widths).

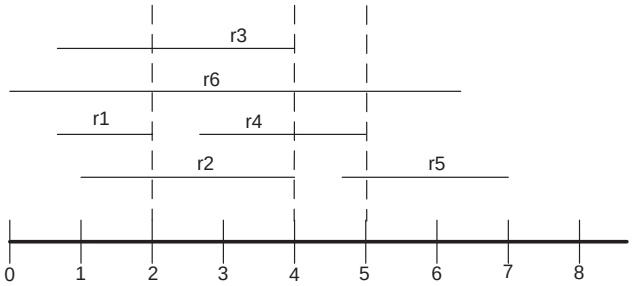
We now present a 5-approximation algorithm for the one-dimensional MWIS problem in a tree. We divide our instances (requests) into two sets: a set consisting of all *narrow* instances, i.e., that have width of at most $1/2$, and a set consisting of all *wide* instances, i.e., that have width greater than $1/2$. 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 uncapacitated case, since no pair of intersecting instances can be simultaneously in the solution. It thus reduces to the problem of finding a MWIS of one-dimensional requests (paths) in a tree which can be solved optimally (see [20]).

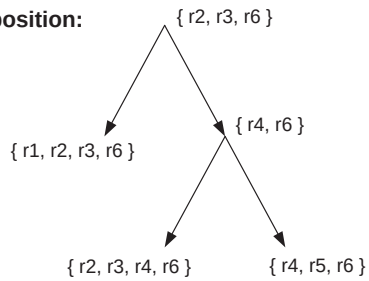
We describe a 4-approximation algorithm for the case all one-dimensional requests are narrow. We begin with some definitions. Let $G = (V, E)$ be a graph with vertex set $V$ and edge set $E$. If $X$ is a subset of the vertices, then $G(X)$ is the subgraph of $G$ induced by $X$. A clique $C$ is called a *separator* if $G(V \setminus C)$ is not connected. An *atom* is a connected graph with no clique separator. Let $C$ be a clique separator of $G$, and let $A, B, C$ be a vertex partition such that no vertex in $A$ is adjacent to a vertex in $B$. Then, $G$ can be decomposed into two components, namely $G' = G(A \cup C)$ and $G'' = G(B \cup C)$. By decomposing $G'$ and $G''$ in the same way, and repeating this process as long as possible, we get a *clique decomposition* of $G$. Figure 4 contains an example of a clique decomposition of a graph. $G$ is thus decomposed into atoms, each being a subgraph of $G$ containing no separator. This decomposition can be represented by a binary tree: each external node represents an atom, and each internal node represents a clique separator. The algorithm described in [20] produces a decomposition tree where the internal nodes lie on one path.

The intersection graph defined by the requests on a tree is the intersection graph of paths on a tree, known as an Edge-Path-Tree (EPT) graph. In order to consider the clique decomposition of an EPT graph [13], we define another class of graphs. A graph is a *line graph* if its vertices correspond to the edges of a multigraph (i.e., a graph with multiple edges) such that two vertices in the original graph are adjacent if and only if the corresponding edges in the multigraph share a common vertex. The following are known results [13]: a graph is a line graph if and only if it is the EPT graph of a star; the atoms of any EPT graph are line graphs. Also, recognizing a line graph and finding its star graph can be done in polynomial time [15].

**Interval Graph:**




**Clique Decomposition:**




We now describe a 4-approximation algorithm for the one-dimensional narrow case, as follows.

1. Given an undirected tree and a set $n$ of paths (instances), define $G$ as the EPT graph of these paths (each vertex in $G$ is a path in the undirected tree).
2. Delete all instances with non-positive profit.
3. If no instances remain, return the empty set. Otherwise, proceed to the next step.
4. Perform the **first** decomposition step of $G$. Let $A, B, C$ be the vertex partition such that $C$ is the clique separator, no edge in $G$ joins a vertex in $A$ and a vertex in $B$, and $G(A \cup C)$ is an atom. Represent the atom $G(A \cup C)$ as a star graph (each vertex in $A \cup C$ is a path in the star), and arbitrarily choose one of the paths in $A$ as $\tilde{I}$.
5. Decompose the profit vector $p$ by $p = p_1 + p_2$.
6. Solve the problem recursively using $p_2$ as the profit function. Let $S'$ be the returned set.
7. 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 $I$. The profit decomposition is the following:

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

**Proposition 1.** *The extended unified algorithm presented above, with* $\alpha = \frac{1}{1-w(\tilde{I})}$, *yields a 4-approximate solution.*

*Proof.* Define $b_{opt}$ to be an upper bound on the optimum $p_1$-profit and $b_{max}$ 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 $\tilde{I} \cup \mathcal{I}(\tilde{I})$ contribute to its $p_1$-profit. As $\tilde{I}$ is a path in a star graph, its path consists of two links. For each of these links $l$, the total width of instances in $\mathcal{I}(\tilde{I})$ that use $l$ in their path is at most 1. In case $\tilde{I}$ is not in the optimal solution, the contribution of the instances in $\mathcal{I}(\tilde{I})$ is at most $2\alpha \cdot p(\tilde{I})$. As $\tilde{I}$ is a path in $A$, it does not intersect with any other path from $B$. However, $\tilde{I}$ intersects with at most two independent groups of instances from $A \cup C$: the instances in one group are paths in the star graph intersecting the first link of $\tilde{I}$, and in the other group the instances are paths in the star graph intersecting the second link of $\tilde{I}$. Otherwise, if $\tilde{I}$ is in the optimal solution, the contribution of the instances in $\mathcal{I}(\tilde{I})$ is at most $2\alpha(1-w(\tilde{I}))\cdot p(\tilde{I})$. Thus, we get $b_{opt} = \max\{2\alpha, 1+2\alpha(1-w(\tilde{I}))\}$.

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

**Corollary 1.** *By choosing the solution with greater profit out of the two sets (wide and narrow), we get an approximation factor of* $1/5$ *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 $R_{12}$. $R_1$ 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 $M$). The proof of the following proposition is omitted.

**Proposition 2.** *The algorithm computes a* $5 \log n$*-approximate solution.*

## Acknowledgements

We would like to thank an anonymous referee for many insightful comments.

## References

1. P.K. Agarwal, M.van Kreveld, ans S. Suri, "Label placement by maximum independent set in rectangles", *Computational Geometry: Theory and applications*, 11(3-4): 209-218 (1998).
2. T. Asano, "Difficulty of the maximum independent set problem on intersection graphs of geometric objects", Proceedings of the Sixth Internat. Conf. on the Theory and Applications of Graphs, Western Michigan University 1988.
3. E. Asplund, B. Grunbaum, "On a coloring problem", Math. Scand. 8 (1960), 181-188.
4. B. Awerbuch, Y. Azar, and S. Plotkin, "Throughput-competitive online routing", Proceedings of the 34th 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 J. on Disc. Mathematics, Vol. 12, pp. 289–297, 1999.
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, Vol. 48 (2001), pp. 1069-1090.
7. R. Bar-Yehuda and S. Even, "A local-ratio theorem for approximating the weighted vertex cover problem", Annals of Discrete Mathematics, Vol. 25, pp. 27–46, 1985.
8. R. Bar-Yehuda, M. Halldorsson, J. Naor, H. Shachnai, and I. Shapira, "Scheduling split intervals", Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, CA, (2002), pp. 732-741.
9. P. Berman, B. DasGupta, S. Muthukrishana, and S. Ramaswami, "Improved approximation algorithms for rectangle tiling and packing", Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, 2001, pp. 427-436.
10. T. Erlebach and K. Jansen, "Off-line and on-line call-scheduling in stars and trees", Proceedings of the 23rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG'97) LNCS 1335, Springer Verlag, 1997, pp. 199-213.
11. 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, 1998, pp. 179-188.
12. T. Erlebach, K. Jansen, and E. Seidel, "Polynomial-time approximation schemes for geometric graphs", Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, 2001, pp. 671-679.

13. M.C. Golumbic and R.E Jamison, "Edge and vertex intersection of paths in a tree", Discrete Math. 55 (1985), 151-159.
14. R. Guerin and A. Orda, "Networks with advance reservations: the routing perspective", Proceedings of INFOCOM 2000, Tel-Aviv, Israel, April 2000.
15. P.G.H Lehot, "An optimal algorithm to detect a line graph and output its root graph", J. ACM 21 (1974) 569-575.
16. W. Reinhardt. Advance resource reservation and its impact on reservation protocols. In *Proceedings of Broadband Islands'95*, Dublin, Ireland, September 1995.
17. J. W. Roberts and K.-Q. Liao. Traffic models for the telecommunication services with advanced reservation. In *Proceedings of the 11th Intl. Teletraffic Congress*, Kyoto, Japan, June 1985. Paper 1.3-2.
18. O. Schelén and S. Pink. Sharing resources through advance reservations agents. In *Proceedings of IWQoS'97*, New York, NY, May 1997.
19. A. Schill, F. Breiter, and S. Kühn. Design and evaluation of an advance resource reservation protocol on top of RSVP. In *Proceedings of IFIP Broadband'98*, Stuttgart, Germany, April 1998.
20. R. Tarjan, "Decomposition by clique separators", Discrete Math. 55 (1985), 221-231.
21. J. Virtamo. A model of reservation systems. *IEEE. Trans. Commun.*, 40(1):109–118, January 1992.
22. D. Wischik and A. Greenberg. Admission control for booking ahead shared resources. In *Proceedings of INFOCOM'98*, San Francisco, CA, April 1998.