Electronic Colloquium on Computational Complexity, Report No. 30 (2006)

Packing to Angles and Sectors

Piotr Berman Jieun Jeong Shiva P. Kasiviswanathan Bhuvan Urgaonkar

Department of Computer Science and Engineering
The Pennsylvania State University
e-mail: {berman,jijeong,kasivisw,bhuvan}@cse.psu.edu

Abstract

In our problem we are given a set of customers, their positions on the plane and their demands. Geometrically, directional antenna with parameters α, ρ, R is a set of points with radial coordinates (θ, r) such that $\alpha \leq \theta \leq \alpha + \rho$ and $r \leq R$. Given a set of possible directional antennas we want to cover all customers positions so that the demands of customers assigned to an antenna stay within a bound. We provide approximation algorithms for three versions of this cover problem.

1 Introduction

During the last decade, we have witnessed a growth in the number and variety of devices that use the wireless communication technology. Until now most papers on wireless networks assumed that the communicating devices were equipped with onmi-directional antennas. However, recently there has been a growing interest in the use of directional antennas. Unlike omni-directional antennas, which transmit in all directions simultaneously, these directional antennas can transmit along a narrow beam in a particular direction. These antennas are able to transmit data to a greater distance by concentrating the radiated power in a narrow angle. Additional benefits that these directional antennas are reported to yield include higher throughput [1, 3], lower interference, and better energy-efficiency [4].

In this paper we study some capacity provisioning problems that arise at a base-station that is using directional antennas to communicate over the wireless network with users geographically spread around it. The users are wireless devices such as laptops, PDAs, etc. We assume a simple model for a directiona antenna in which it transmits over a sector of a circle. The antenna can increase the angle of the sector by reducing its radius and vice versa - the exact relation between the angle and the radius does not affect our analysis. Each directional antenna is assumed to have a total bandwidth (capacity) B. The base-station *assigns* a directional antenna to each user u. This involves ensuring the following: (i) the user u lies within the sector covered by the antenna and (ii) the antenna reserves some bandwidth $b(u)$ $(0 \leq b(u) \leq B)$ for the user u such that the total bandwidth reserved by the antenna does not exceed B.

In the classic bin packing problem (see [2] for a survey) we are given a set of objects $\{1, \dots, n\}$, each with a *weight*, d_i, and the goal is to have a partition into as few parts as possible such that elements of each part can be placed in a single bin, *i.e.* that have weights not exceeding some fixed bin capacity.

ISSN 1433-8092

Similar problems, but with additional limitations concerning the partition, arise in many applications. Bansal, Chan *et al.* considered recently one possible limitation, defining *fragility* of objects. Limitations considered in this paper require elements of a part to fit within a geometrically defined enclosure, like angle, interval or a sector.

The enclosures considered in this paper are related to possible ranges of directional antennas. In this setting we have a mast that is connected to a satellite or a fiber-optic cable, and we want to transfer information to and from a number of customers that can be reached by antennas placed on this mast. The customers may have different demands on the transmission capacity, e.g. they may subscribe to broad-band connections with various bandwidth. The customers have fixed positions and we can choose how to direct our antennas.

In MinAntVar problem we can choose the angular range of an antenna, so when the range is narrower it can reach further, we have a limit on the total bandwidth demand that can be assigned to an antenna and we want to minimize the number of antennas. We show an approximation algorithm for MinAntVar with ratio 3.

If the angular range of antennas are always fixed, the problem becomes basically 1-dimensional. We minimize the number of antennas, problem MinAnt, and for a fixed number of antennas we minimize the maximum demand assigned to a single antenna, problem MinAntLoad. For both problems we present an approximation algorithm with approximation ratio 1.5.

Our approximation results are provably tight for MinAnt problem, because approximating it within a ratio better than 1.5 (even asymptotically) would imply solving Partition problem in polynomial time. We can place a group of customers very close to each other with demands that form an instance of Partition: covering that group with 2 antenna sets means finding the positive answer to Partition; we can replicate this group m times, so that no antenna set contains customers from different groups. As a result, if for every case that can be solved with $2m$ antennas we can find a solution with fewer than $3m$ antennas we can solve Partition.

Our packing problem has another motivation. Suppose that we are sending some goods that were ordered, and each order has weight d_i, arrival time t_i and it has to be shipped at latest at time $t_i + p_i$ (say that p_i is the *patience* of a customer). Given a capacity of a single shipment (truckload, container, etc.) we want to minimize the number of shipment.

If every p_i is equal to some p, then we have a problem that is essentially identical to MinAnt; shipment at time t can be used for customers with $t - p \leq t_i \leq t$. The only difference is that time is not circular, which makes inductive arguments easier. If the values of p_i's are arbitrary, we can easily adapt our algorithm for MinAntVar to provide approximate solutions for this problem. As a result, we can approximate this problem within factor 1.5 in the case of constant patience parameter and within factor 3 in the case of variable patience parameters.

2 Preliminaries

We will deal with points in radial coordinates and with circular orderings, and for the sake of clarity we introduce several definitions.

A point (θ, r) is equivalent to a Cartesian point $r \times \sin\theta, r \times \cos\theta$. Equality of angles is understood modulo 2π, *i.e.* $\alpha = \beta$ means that for a certain integer i we have

define $\mathsf{LeftR}(i) = R^{rev}(\theta_i, r_i)$
define $\mathsf{RightR}(i) = R(\theta_i, r_i)$
define $\mathsf{DoubleR}(i) = \mathsf{LeftR}(i) \cup \mathsf{RightR}(i)$

$V \leftarrow \{1, \ldots, n\}$
$\mathcal{C} \leftarrow I \leftarrow \emptyset$
while $\mathcal{C} \neq V$
  $i \leftarrow$ element of $V - C$ with maximum r_i
  $I \leftarrow I \cup \{i\}$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathsf{DoubleR}(i)$

Figure 1: Phase One of the algorithm for variable antennas

$\alpha = \beta + 2i\pi$. For $\lambda \leq 2\pi$ we define $\mathsf{Sector}(\alpha, \lambda) = \{(\theta, r) : \exists\gamma\ 0 \leq \gamma \leq \lambda$ and $\theta = \alpha + \gamma\}$.

We also define $\mathsf{Disk}(r) = \{(\theta, s) : s \leq r\}$. A sequence of angles $\alpha_1, \ldots, \alpha_k$ is circularly ordered if for a certain β we have $0 \leq \alpha_1 + \beta \leq \cdots \leq \alpha_k + \beta < 2\pi$.

In each problem under consideration, our data contains a set of customers $U = \{1, \ldots, n\}$, customer i is located at point (θ_i, r_i) and it has demand $d_i > 0$. It will be convenient to use notation $L = \{i \in U : d_i > 1/2\}$ for the set of customers with *large* demands.

3 The Case of Variable Range of Antennas

3.1 Brief Problem Statement

There exists a decreasing function $\rho(r)$ that describes the trade-off between the radius of the antenna range and its angular width. For a set of customers S we define its demand as $d(S) = \sum_{i \in S} d_i$. An antenna has a range

$$R(\alpha, r) = \{i : (\theta_i, r_i) \in \mathsf{Sector}(\alpha, \rho) \cap \mathsf{Disk}(r)\}$$

We will also use a shorthand: $R^{rev}(\alpha, r) = R(\alpha - \rho, r)$; intuitively the sectors of $R(\alpha, r)$ and $R^{rev}(\alpha, r)$ both start at α and then they extend in opposite directions. A valid solution to MinAntVar problem is a partition $U = S_1 \cup \cdots \cup S_k$ such that for $j = 1, \ldots, k$ we have

- $d(S_j) \leq 1$;
- S_j is contained in an antenna range $R(\alpha_j, s_j)$.

We want to minimize k, which is the number of different antennas we deploy to satisfy demands of all customers.

3.2 Phase 1 of the Algorithm — Independent Set

We start by forming a cover of the set of all customers with antenna ranges, we will also create an *independent* set of customers I; more precisely, we will assure that no two different customers in I may belong to the same antenna range. The method is described in Fig. 1.

The ranges $\mathsf{LeftR}(i)$ and $\mathsf{RightR}(i)$ are defined in such a way that if some customer j has $r_j \leq r_i$ and i and j can be together in an antenna range, then $j \in \mathsf{LeftR}(i) \cup$

$\mathsf{RightR}(i) = \mathsf{DoubleR}(i)$. Thus it is easy to see that I is indeed an independent set of customers, so we need at least $|I|$ antenna ranges to cover all customers, while we have found a cover that uses $2|I|$ antenna ranges.

It will be convenient to duplicate customers in set I, so for $i \in cI$ we create customer $n+i$ with $(\theta_{n+i}, r_{n+i}, d_{n+i}) = (\theta_i, r_i, 0)$. Then we make a substitution to make V and I disjoint:

$$I \leftarrow \{n+i : \ i \in I\}$$

3.3 Phase 2 of the Algorithm — Maximum Flow

We define a network as follows: there are 4 layers of nodes:

layer 0:	r,	the source,
layer 1:	V,	the set of Demand Customers,
layer 2:	I,	the Independent Customers,
layer 3:	s,	the sink.

Arcs have capacities defined as follows:

(r, i) always exists,	$u_{r,i} = d_i$,
(i, j) exists iff $i \in \mathsf{DoubleR}(j)$,	$u_{i,j} = d_i$
(j, s) always exists,	$u_{j,s} = 1$.

In this network we find a maximum flow f and a cut C with the minimum capacity. We normalize f and C to assure certain desirable properties that will be used to define the partition of the customer and to analyze the approximation ratio.

Let $\mathsf{Partial}$ be the set of undirected edges $\{a, b\}$ such that for a network arc (a, b) we have $0 < f_{a,b} < u_{a,b}$. Our first normalization criterion is that $\mathsf{Partial}$ is acyclic.

Lemma 1 *Given a valid flow f we can in polynomial time find flow f' such that (a) values of f' and f are the same, (b) $\mathsf{Partial}$ defined by f' is acyclic, (c) if $f_e = 0$ or $f_e = u_e$ then $f_e = f'_e$.*

Proof. If $\{a, b\} \in \mathsf{Partial}$ (defined by f) then in the residual network defined by f it corresponds to two arcs with positive capacity, *e.g.* (a, b) has capacity $u_{a,b} - f_{a,b}$ and (b, a) has capacity $f_{a,b}$. If we have a cycle in $\mathsf{Partial}$, then we also have a directed cycle in the residual network with positive capacities on arcs, and if the minimum of these capacities is c, we can augment f by adding c to $f_{a,b}$ if (a, b) is on the directed cycle, and by subtracting c from $f_{a,b}$ if (b, a) is on the directed cycle. Afterward any arc on the cycle that had residual capacity c does not correspond to an edge in $\mathsf{Partial}$ anymore. We can repeat the process until $\mathsf{Partial}$ is empty. Note that the augmentations change flow values only in arcs in which the flow is neither 0 nor the maximum possible. ❑

The process described by Lemma 1 will be called *acyclic normalization*. We will say that f is *bad* if for some small demand g and large demand h and independent customer j we have $0 < f_{g,j} < d_g$, $f_{r,h} = 0$, while arc (g, j) exists. Our second normalization is that f cannot be bad.

Lemma 2 *Given a valid flow f we can in polynomial time find flow f' such that (a) values of f' and f are the same, (b) $\mathsf{Partial}$ is acyclic, (c) f' is not bad.*

Proof. We will describe possible steps of the normalization process, to estimate their number we will introduce two quantities, ρ and β, that we will use as counters.

$$\begin{aligned}\rho(f) &= |\{i \in V : d_i \leq 1/2 \text{ and } f_{r,i} = d_i\}| \\ \beta(f) &= |\{(i,j) : d_i > 1/2 \text{ and } 0 < f_{i,j} < d_i\}|\end{aligned}$$

If flow f is bad we can modify it in such a way that $\rho(f) - \beta(f)$ increases.Suppose that f is bad because of g, h and j.
Case 1: $f_{g,j} = c \leq d_h$; we increase $f_{r,h}$ and $f_{h,j}$ from 0 to c, we decrease $f_{r,g}$ by c and we decrease $f_{g,j}$ to 0. We do not decrease $\rho(f)$ and we decreased $\beta(f)$. Subsequent acyclic normalization does not decrease $\rho(f) - \beta(f)$.
Case 2: $f_{g,j} > d_h$, we increase $f_{r,h}$ and $f_{h,j}$ from 0 to d_k (this increases $\rho(f)$) and we decrease $f_{r,g}$ and $f_{g,j}$ by d_k. ❑

Now we normalize the cut C. If arc (i,j) exists and $j \notin C$, then we insist that $i \notin C$. Otherwise we remove i from C, arcs of the form (i,j') cannot contribute to the capacity of C anymore, and at least one did, while now (r,i) makes the contribution of d_i to the capacity of C, clearly, we can improve our criterion (decrease the number of arcs that violate it) with no increase to the capacity of C.

3.4 Phase 3 of the algorithm — Assignment of Demands

Rule 1: If $f_{i,j} = d_i$, we assign customer i to independent customer j.

To formulate the second rule, we form bipartite graph (U, I, A) where A are arcs (i,j) such that $\{i,j\} \in \mathsf{Partial}$, *i.e.* such that $0 < f_{i,j} < d_i$; part one, $U \subseteq V$ consists of starting points of arcs in A. This graph satisfies

Lemma 3 *Bipartite graph* (U, I, A) *contains a matching of size* $|U|$.

Proof. We will prove that (U, I, A) satisfies the assumption of Hall theorem. For a set $X \subseteq U$ we define the set of neighbors of X, $N(X) = \{w : \exists x \in X \text{ s.t. } (x,w) \in A\}$. The assumption of Hall theorem states that $|N(X)| \geq |X|$.

We can break $X \cup N(X)$ into weakly connected components $X_1 \cup N(X_1), \ldots, X_k \cup N(X_k)$, then it suffices to prove the inequality for every X_ℓ. Thus we may assume that $(X, N(X), A)$ is weakly connected.

Now, it suffices to show that X is incident to at least $2|X| - 1$ arcs of A; would it have $|X| - 1$ neighbors or less we would have a subgraph of $\mathsf{Partial}$ with at least $2|X| - 1$ edges and at most $2|X| - 1$ nodes, a contradiction because this implies a cycle in $\mathsf{Partial}$.

So suppose that $(X, N(X), A)$ is weakly connected and the number of arcs incident to X, $\sum_{x \in X} |N(\{x\}| < |X| - 1$. Because each term in this summation is at least 1, this implies that at least 2 of them are 1. Suppose then that for $x \neq y$, $x, y \in X$ we have i and j such that $N(\{x\}) = \{i\}$ and $N(\{y\}) = \{j\}$. Then $f_{r,x} = f_{x,i} < d_x$ and $f_{r,y} = f_{y,j} < d_y$. Thus arcs of A show one path in $\mathsf{Partial}$ from x to y, while we have another, (x, r, y), and so we have a cycle in $\mathsf{Partial}$, a contradiction. ❑

Let M be the matching that exists according to Lemma 3.
Rule 2: If $(i,j) \in M$ we assign customer i to independent customer j.

If $f_{r,i} > 0$, then customer i is assigned to an independent customer, either by Rule 1 or by Rule 2. Thus it remains to handle cases when $f_{r,i} = 0$.

Rule 3: If $f_{r,i} = 0$, then customer i is assigned to an arbitrary independent customer j such that $(i, j) \in A$ (equivalently, such that $i \in \mathsf{DoubleR}(j)$).

Once a customer is assigned to independent customer j, if $\theta_i < \theta_j$ then j assigns i to $\mathsf{LeftR}(j)$ and otherwise j assigns i to $\mathsf{RightR}(j)$.

When the sum of demands assign to a single antenna range exceeds 1 we introduce more antennas with exactly the same range. At this point we have instances of the bin-packing problem. In the analysis it is important that we pack demands according to Rule 1 first, then according to Rule 2 and than according to Rule 3 and we try to fit them in already open bins (so-called Any Fit rule).

3.5 Assignment of Potentials and the Analysis

Now we will utilize the minimum cut C and its normal property.

Classification of Customers: for i in Layer 1, if $i \in C$ we put i into the class of IntenseRelaxed Customers, and if $i \notin C$, we put i into the class of Intense Customers. Note that if i is Relaxed, then arc (r, i) contributes to the capacity of C and thus $f_{r,i} = d_i$. A Relaxed customer may have a smaller flow.

Classification of Independents: for j in Layer 2, if $j \in C$ we put j into the class of Intense Independents, and if $j \notin C$ we put j into the class of Relaxed Independents. Again, if j is Intense, then $f_{j,s} = 1$, and if j is Relaxed, it may have a smaller flow.

Because C is normalized, we have no arcs from Relaxed Customers to Relaxed Independents. Also, because our flow had the value equal to the capacity of C, each path flow was crossing exactly one edge contributing to the capacity of C, as a result we have no assignments (marked by positive flow) of Intense Customers to Intense Independents.

Initial distribution of potential: a Relaxed Customer i has potential $2d_i/3$, and an Intense Customer i has potential $d_i/3$. Similarly, a Relaxed Independent has potential $2/3$ and an Intense Independent has potential $1/3$.

The first observation is that if S is a part from a partition that forms a valid solution, then the sum of potentials of the elements of S does not exceed 1. If S contains no Independent, then its potential comes only from the demands of Customers that adds to at most 1 and the respective potentials add to at most $2/3$. If S contains an Intense Independent, we increase this estimate by $1/3$, but this is still at most 1. If S contains a Relaxed Independent, with potential of $2/3$, then all of its Customers are Intense, hence their demands contribute at most $1/3$ to the potential of S. This proves that the size of the optimum partition is at least as large as the sum of all potentials.

Now it suffices to show that to every antenna we introduce we can assign at least potential of $1/3$.

Second distribution of potential from Intense Customers An Intense Customer has assignments to Relaxed Independents only. We give its potential to the Relaxed Independent to which it was assigned (alway by Rule 1 or Rule 2).

Balance of a Relaxed Independent j If j collects no more than $1/3$ of the potential from the assigned Customers, it does not have to create duplicate antennas, so it has to justify two antennas only, and for that its own potential of $2/3$ is adequate. Otherwise, it received total demand exceeding 1, and thus an extra $1/3$ of potential, enough for a single duplicate antenna. Note that all assignments according to Rule 1 fit in one "bin", and that the single assignment according to Rule 2 has to fit in one "bin" as well, hence one can always pack them in two bins, and thus a Relaxed Independent has to justify at most 3 antennas.

Second distribution of potential from Relaxed Customers A Relaxed Customer has assignments to Intense Independents only. If it was assigned according to Rule 1 and Rule 2, we collect its potential to a "central location"; because their sum is at least equal to the number of Intense Independents, (it suffice to saturate the flow from the Intense Independents to s), we can assign potential 2/3 from the central location to each Intense Independent. If a Relaxed Customer i was assigned to j according to Rule 3, it brings its potential to j.

Balance of an Intense Independent j

Because of the Second Distribution, every Intense Independent has potential 1 before we start applying Rule 3.

Case 1: Rule 2 assigned to j some i with $d_i > 1/2$. Independent j has two ranges that accommodate all demands assigned by Rule 1, and one range to accommodate i, and potential 1 to justify these three ranges. Because f is not bad, every k that can be assigned to i by Rule 3 has $d_k > 1/2$, and thus it brings potential of $2d_k/3 > 1/3$ that suffices to justify a new antenna.

Case 2: Rule 2 assigned to j some i with $d_i \leq 1/2$ (if it assigned nothing, pretend that it assigned some j with $d_j = 0$). Without loss of generality, Rule 2 assigns demand a to $\mathsf{LeftR}(j)$, while Rule 1 assigned b to $\mathsf{LeftR}(j)$ and c to $\mathsf{RightR}(j)$. Note that $a \leq 1/2$ and $b + c \leq 1$.

Case 2.1: $c \leq 1/2$. Independent j creates a copy of $\mathsf{LeftR}(j)$ to accommodate $a \leq 1/2$ alone. Later, when Rule 3 is applied, Any Fit creates duplicates of $\mathsf{LeftR}(j)$, and at most one can be loaded with less than $1/2$ of demand, say with x of demand. In this case, Any Fit had to increase the load of the copy that accommodated a to $1-x$, thus by at least $1/2-x$, and thus the copy that got only x can get also $1/2-x$. The same argument applies to the copies of $\mathsf{RightR}(j)$ created by Any Fit, because we start with a copy loaded with $c \leq 1/2$.

Case 2.2: $c > 1/2$. Note that $a + b + c \leq 3/2$. Independent j packs $a + b$ together, and thus before Rule 3 is applied it has a spare capacity of x in $\mathsf{LeftR}(j)$, spare capacity of y in $\mathsf{RightR}(j)$ where $x + y \geq 1/2$, plus spare potential of $1/3$.

Any Fit creates copies of the $\mathsf{LeftR}(j)$ and adds some ε to the load of the copy that initially had spare capacity x, as a result it creates a number of copies loaded with at least $1/2$ of potential and possibly a copy with only δ. If the latter happens, we use $\varepsilon + \delta$ to justify the last copy. Because $1 - x + \varepsilon + \delta > 1$, we have $\varepsilon + \delta > x$, and thus we may have a temporary deficit of potential of the size below $2/3(1/2 - x)$.

Similar reasoning applied to the activities of Any Fit with copies of $\mathsf{RightR}(j)$ shows that a temporary potential deficit may be created, but no more than $2/3(1/2-y)$. The sum of the two deficits is thus at most $2/3(1 - x - y) \leq 2/3 \times 1/2 = 1/3$, and we have $1/3$ of spare potential.

3.6 Conclusion

We have shown an algorithm that runs in polynomial time, its most time consuming phase is finding the maximum flow. Subsequent normalization amounts to no more than a standard augmentation process that is a part of a maximum flow algorithm to begin with. Similarly, finding a maximum matching can be done by finding a maximum flow in a very similar network.

Our potential analysis shows that if the sum of potentials is P, any solution must use at least P antennas and we use at most $3P$. We may conclude with the following

Theorem 4 *There exists a polynomial time approximation algorithm for the problem of capacitated cover with variable antennas with ratio* 3.

4 Fixed Range Antennas

4.1 Brief Problem Statements

Our reasoning could be a bit simpler if we could define linear order of all customers such that allowed antenna ranges are intervals in that order. However, even though we cannot introduce a consistent global order we can define local ordering which is good enough for our purposes.

Without loss of generality we assume that θ_i is different for each $i \in U$. Let $R(i) = \{j \in U : \theta_j = \theta_i + \beta$ and $0 \le \beta \le \rho$ for some $\beta\}$ and $R^{rev}(i) = \{j \in U : \theta_j = \theta_i - \beta$ and $0 \le \beta \le \rho$ for some $\beta\}$.

Given that, we can formulate two problems to which a valid solution is a partition $U = S_1 \cup \cdots \cup S_K$ such that $S_i \subseteq R(j_i)$ for $i = 1, \ldots, K$. In MinAnt problem we require that $d(S_i) \le 1$ and we minimize K. In MinAntLoad problem we fix K and we minimize $\max d(S_i)$ (note that solving the cover problem allows to find out when K is too small to give any feasible solution).

We will show approximations for both problems with ratio 1.5. We reformulate MinAntLoad as follows: we are given input describing the set of customers and the angle ρ, as well as D, the minimum value of the maximum demand of a set in a legal solution with K sets.

Given that, we have to find a legal solution in which the maximum demand of a set is at most $1.5D$. Because the number of different D's that result in different runs of our algorithm is small, we can efficiently obtain a result that is as good as if we would try all of them.

In that formulation we can rescale the customer demands so we get $D = 1$. In this preliminary version, we can omit the case when $\rho \ge \pi$. This case is very close to bin packing and it does not pose any challenge, but it requires a tedious case analysis.

4.2 Relaxed Solutions, Normalization

We define graph $(U, Succ)$ where $(i,j) \in Succ \equiv R(i) \cap R^{rev}(j) = \{i,j\}$. Note that if i and j are in distinct weakly connected components of $(U, Succ)$ then they connot belong to the same $R(i)$; as a result we can connect components into a single path and it is still the case that every $R(i)$ forms a path in $(U, Succ)$. Therefore we can assume that $(U, Succ)$ forms simple cycle.

We can remove L or $U - L$ from the cycle of $(U, Succ)$ to obtain graphs $(L, Succ_L)$ and $(U - L, Succ_s)$. If $(i,j) \in Succ$ (respectively, $Succ_L$, $Succ_s$), we will use notation $j = Succ(i)$ (respectively, $j = Succ_L(i)$, $j = Succ_s(i)$).

A relaxed set S is defined by variables $x_{S,i}$ for each $i \in U$ that describe to what degree i belongs to S. A normal set T can be viewed as a special case of a relaxed set, such that $x_{T,i} \in \{0, 1\}$. We will use this notation for L and $U - L$.

We define

$$\begin{aligned}
\widehat{S} &= \{i \in U : x_{S,i} > 0\} \\
\mathit{Segment}(S) &= R(i) \cap R^{rev}(j) \text{ if } i, j \in \widehat{S} \text{ and } \widehat{S} \subset R(i) \cap R^{rev}(j) \\
\mathit{segment}(S) &= \mathit{Segment}(S - L)
\end{aligned}$$

We extend notation $S - L$ and $S \cap L$ to relaxed sets with this formula:

$$x_{S-L,i} = x_{U-L,i} x_{S,i}, \qquad x_{S \cap L,i} = x_L x_{S,i}$$

A collection $\mathcal{F}$ of relaxed sets forms a relaxed solution to MinAnt or MinAntLoad if

$$\mathit{Segment}(S) \text{ is defined for each } S \in \mathcal{F} \tag{1}$$

$$\sum_{S \in \mathcal{F}} x_{S,i} = 1 \text{ for each } i \in U \tag{2}$$

$$d(S) = \sum_{i \in U} d_i x_{S,i} \leq 1 \text{ for each } S \in \mathcal{F} \tag{3}$$

A relaxed set S is *good* if $0 < x_{S,i} < 1$ implies that $\mathit{segment}(S) = R(j) \cap R^{rev}(k)$ and $i \in \{j, k\}$. Note that if S is a good relaxed set, $x_{S,i} > 0$ may be true for at most one $i \in L$, and if it is true, $x_{S,i} = 1$. In this case we use notation $i = L(S)$.

Lemma 5 *Any solution F to an instance of* MinAnt *can be replaced with a relaxed solution $\mathcal{F}$ in which each $S \in F$ is replaced with a good relaxed set $S' \in \mathcal{F}$ such that (i) $S' \cap L = S \cap L$, (ii) $d(S' - L) = d(S - L)$, and (iii) $\mathit{segment}(S') \subset \mathit{segment}(S)$.*

Proof. Consider $S \in F$, if $S \subset L$ then we set $S' = S$. Let F_0 be the family of the remaining sets of F. For $S \in F_0$ we define $\mathit{first\_small}(S)$ as such j that for some k we have $\mathit{segment}(S) = R(j) \cap R^{rev}(k)$. We order F_0 as $S_1, S_2, \ldots, S_m$ so that in (U, Succ) has a simple path with subsequence $\mathit{first\_small}(S_1), \mathit{first\_small}(S_2), \ldots, \mathit{first\_small}(S_m)$. The process of replacing each S_i with S'_i can be described as an algorithm in Fig. 2. Properties (i) and (ii) follow straightforwardly from the algorithm, and so are the properties (2) and (3) of a valid relaxed solution. It remains to show that for some initial value of D we assure property (iii) and thus property (1) of a valid relaxed solution.

For $S \in F$ define

$$\begin{aligned}
\mathit{Debt}(S, T) &= \begin{cases} \varnothing \text{ if } \mathit{first_small}(S) \notin \mathit{segment}(T) \\ \mathit{segment}(S) \cap T \cap (U - L) \text{ otherwise} \end{cases} \\
\mathit{Debt}(S) &= \bigcup_{T \in F} \mathit{Debt}(S, T) \\
\mathit{debt}(S) &= d(\mathit{Debt}(S))
\end{aligned}$$

With respect to a run of our algorithm, we also define $\mathit{offset}(S_k)$: let $(i_1, \ldots, i_\ell)$ be the path in $(U - L, \mathit{Succ}_s)$ that connects elements of $\mathit{segment}(S_k) - L$, and assume that when the algorithm sets j to k we have $i = i_a$ and $r = r_0$; then $\mathit{offset}(S_k) = r_0 + \sum_{h=1}^{a-1} d_h$.

It is easy to show that $\mathit{offset}(S_j) = \mathit{debt}(S_j)$ implies $\mathit{offset}(S_{j+1}) = \mathit{debt}(S_{j+1})$. Thus if we set the initial value of D to $\mathit{debt}(S_1)$ then we have $\mathit{debt}(S_j) = \mathit{offset}(S_j)$ for $j = 1, \ldots, m$.

```
j ← 0
i ← first_small(S_1)
r ← d_i
do forever
    if D > r
        x_{S_j,i} ← r/d_i
        D ← D − r
        i ← Succ_s(i)
        r ← d_i
    else
        x_{S_j,i} ← D/d_i
        if j = m
            terminate
        r ← r − D
        j ← j + 1
        D ← d(S_j − L)
```

Figure 2: Algorithm converting sets $S_1, \ldots, S_m$ in a solution to good relaxed sets. The initial value of D is described in the proof, variables $x_{S_0,i}$ do not correspond to any of the constructed relaxed sets.

One can also see that if $\mathtt{offset}(S_j) \geq 0$ and $\mathtt{offset}(S_j)+d(S_j-L) \leq d(\mathtt{segment}(S_j)-L)$ then $\mathtt{segment}(S') \subset \mathtt{segment}(S)$.

Finally, $\mathtt{debt}(S_j) + d(S_j - L) \leq d(\mathtt{segment}(S_j) - L)$ because $\mathtt{Debt}(S_j)$ and $S_j - L$ are two disjoints subsets of $\mathtt{segment}(S_j)$. ❑

Below we will use term *good solution* to describe a relaxed solution that contains only good relaxed sets. Note that if $\mathtt{Segment}(S) = \{i\}$ we can set $x_{S,i} = 1$, and set $x_{T,i} = 0$ for $T \neq S$ and we still have a good solution.

We remove from consideration every S such that $\mathtt{Segment}(S) = \{L(S)\}$. To describe how to convert a good solution onto an approximate solution we first define succession of sets that have non-empty $S - L$. For $i \in U - L$ let $\mathcal{F}_i' = \{S \in \mathcal{F} : \ \mathtt{segment}(S) = \{i\}\}$. If $|\mathcal{F}_i| = 1$ then the question of ordering $\mathcal{F}_i$ is moot. Otherwise $0 < x_{S,i} < 1$ for each $S \in \mathcal{F}_i$ and hence $L(S)$ is defined. There exists a path (that is not a cycle) in $(U, \mathtt{Succ})$ that contains every such $L(S)$ and we determine the succession according to the precedence of $L(S)$'s on this path. Now let assume that (i, j, k) is a path in $(U - L, \mathtt{Succ}_s)$ and let $\mathcal{F}_j = \{S \in \mathcal{F} : \ x_{S,j} > 0\}$. Note that $x_{S,i} > 0$ is true for at most one S in $\mathcal{F}_j$, if such a set exists, it starts the ordering of $\mathcal{F}_j$; similarly if $x_{S,k} >$ then S ends the ordering of $\mathcal{F}_j$, the rest of this ordering is the ordering of $\mathcal{F}_j'$. Finally, if $\mathcal{F}_i$ and $\mathcal{F}_j$ are disjoint, we have a succession from the last set of $\mathcal{F}_i$ to the first set of $\mathcal{F}_j$.

To round $\mathcal{F}$ we split it into paths of the succession order where successive sets S, T are connected if $\mathtt{segment}(S) \cap \mathtt{segment}(T) = \{i\}$ for some i, denote i as junction $J(S, T)$. A path has $k+1$ sets and k junctions. To get a solution of MinAnt, we create sets that contain consecutive pairs of junctions. If we have $2\ell+1$ good sets, add ℓ new sets, and if we have 2ℓ good sets, we add $\ell - 1$ sets with two junctions each and 1 set with just one junction. If our path of sets forms a full cycle, we first add each junction to the preceding set, and at least one set will still have $d(S) \leq 1$; we remove this set from consideration and we handle the remaining sets.

It is simpler to obtain a solution of MinAntLoad in which every $d(S)$ is at most 1.5: we allocate each junction to its preceding set. To find a minimum size good solution we will use dynamic programming. However, to assure that we do not have to consider more than a polynomial number of possibilities we have to limit the allowed solution even further.

We will define a symmetric relation on good sets "S crosses T". This relation holds only if $\mathrm{Segment}(S) \cap \mathrm{Segment}(T) \neq \emptyset$. In that case, we use notation $\mathrm{path}(S,T) = (i_1, \ldots, i_\ell)$ for the path in (U, Succ) that connects all elements of $\mathrm{Segment}(S) \cup \mathrm{Segment}(T)$ (we may also use notation $\mathrm{path}(S)$). S does cross T if $\mathrm{Segment}(S) \subset \mathrm{Segment}(T)$. Moreover, S crosses T if for some $a < b$ and $c < d$ we have $L(S) = i_a$, $L(T) = i_b$, $i_c \in \mathrm{segment}(T)$ and $i_d \in \mathrm{segment}(S)$. Lastly, if S crosses T and $\{i_1, i_\ell\} \not\subset U - L$ we say that S strongly crosses T.

Given a family $\mathcal{F}$ of relaxed sets, we consider the graph $(\mathcal{F}, \mathrm{crosses})$. Our goal is to limit the connected components of this graph to a very special form.

Lemma 6 *If relaxed set* $S \in \mathcal{F}$ *strongly crosses* $T \in \mathcal{F}$ *then either* $\mathrm{segment}(S) \subset \mathrm{Segment}(T)$ *or* $\mathrm{segment}(T) \subset \mathrm{Segment}(S)$.

Proof. Using the above notation, we can assume that $i_\ell = L(T)$. If $\mathrm{Segment}(S) \subset \mathrm{Segment}(T)$ then clearly $\mathrm{segment}(S) \subset \mathrm{Segment}(T)$ and the claim is true. Otherwise, $L(S) = i_a$ for some $a < \ell$ and therefore for some c, d we have $c < d$, $i_c \in \mathrm{segment}(T)$ and $i_d \in \mathrm{segment}(S)$. Because S is a good set, $\mathrm{segment}(S) \subset \{i_c, i_{c+1}, \ldots, i_\ell\} \subset \mathrm{Segment}(T)$. ❑

Remark. In the above proof we had the implication: if S crosses T and the first or the last element on $\mathrm{path}(S, L)$ equals $L(T)$, then $\mathrm{segment}(S) \subset \mathrm{Segment}(T)$.

Lemma 7 *Assume that* $R, S, T \in \mathcal{F}$*,* S *strongly crosses* T*,* $\mathrm{path}(S,T) = (i_1, \ldots, i_\ell)$ *and that for some* a, b, c *we have* $1 \leq a < b < c \leq \ell$*,* $i_a \in \mathrm{segment}(S)$*,* $i_b \in \mathrm{segment}(R)$*,* $i_c \in \mathrm{segment}(T)$*. Then* R *strongly crosses* S *or* T*.*

Proof. Because of symmetry, and by Lemma 6, we can assume that $\mathrm{segment}(S) \subset \mathrm{Segment}(T)$. This implies that $L(T) = i_1$ and because R is a good relaxed set, $\mathrm{segment}(R) \subset \mathrm{Segment}(T)$. If $\mathrm{Segment}(R) \subset \mathrm{Segment}(T)$ then R strongly crosses T, so we can assume that $L(R) \notin \mathrm{Segment}(T)$. We can now consider a path in (U, Succ) that connects all elements of $\mathrm{Segment}(S) \cup \mathrm{Segment}(T) \cup \{L(R)\}$. If $L(R)$ follows i_1 on that path then again R strongly crosses T. If $L(R)$ precedes i_1, then $\mathrm{segment}(S) \subset \mathrm{Segment}(R)$; if $\mathrm{Segment}(S) \subset \mathrm{Segment}(R)$ then S strongly crosses R, otherwise $L(S)$ follows $L(S)$ while $i_a \in \mathrm{segment}(S)$ precedes $i_b \in \mathrm{segment}(R)$, so once more, R strongly crosses S. ❑

Lemma 8 *A good solution* $\mathcal{F}$ *to* MinAnt *can be replaced with another solution that has the same number of relaxed sets and in which no set strongly crosses another set.*

Proof. For a relaxed set S we define $\mathrm{length}(S)$ as follows: if $\mathrm{Segment}(S) = \{i\}$ then $\mathrm{length}(S) = x_{S,i}$; if $|\mathrm{Segment}(S)| \geq 2$, we can assume that $\mathrm{Segment}(S)$ is connected in (U, Succ) with a path $(i, \ldots, j)$, and we define $\mathrm{length}(S) = |\mathrm{Segment}(S) - \{i,j\}| + x_{S,i} + x_{S,j}$. We will use the following characteristic of a relaxed solution: we place

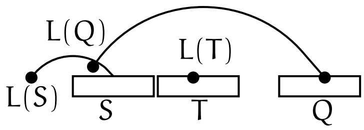

Figure 3: Illustration for the proof of Lemma 9.

$\mathsf{length}(S)$ for $S \in \mathcal{F}$ in non-increasing order that we call *ordered length vector*. We will show that if $\mathcal{F}$ contains some S, T that strongly cross each other then we can modify $\mathcal{F}$ in such a way that the new ordered length vector will lexicographically preceded the former one.

Suppose that $S, T \in \mathcal{F}$ and S strongly crosses T and $\mathsf{path}(S, T) = (i_1, \ldots, i_\ell)$. If there exist $R \in \mathcal{F} - \{S, T\}$ and $a < b < c$ such that $\{i_a, i_c\} \subset \mathsf{segment}(S) \cup \mathsf{segment}(T)$ while $i_b \in \mathsf{segment}(R)$ we replace S, T with a strongly intersecting pair R, S or R, T. We repeat it until something like that does not happen. Define relaxed set $Q = (S - L) + (T - L)$, i.e. $x_{Q,i} = x_{S-L,i} + x_{T-L,i}$. In $(U - L, Succ_s)$ set $\widehat{Q}$ is connected with a path $(j_1, \ldots, j_m)$ where $m > 1$ (otherwise S would not cross T), and because of our choice of S and T, we have $x_{Q,j_a} = 1$ for $a = 2, \ldots, m-1$.

By symmetry, we can assume that $\{i_1\} = T \cap L$ and thus, by Lemma 6, $\mathsf{segment}(S) \subset \mathsf{Segment}(T)$. This means that for some a we have $\mathsf{segment}(S) = \{j_1, \ldots, j_a\}$ and $\mathsf{segment}(T) \subseteq \{j_a, \ldots, j_m\}$. We will alter S and T using the algorithm from Fig. 2 with: $S_1 = T$, $S_2 = S$, i initialized to j_1 and D initialized to $x_{Q,j_1} d_{j_1}$. In other words, the initial part of Q was in S and the final part of Q was in T and now we reverse that order, while leaving $d(S - L)$ and $d(T - L)$ unchanged.

One can see that new $\mathsf{Segment}(T)$ is a subset of the former, and that $\mathsf{length}(T)$ decreased: either new x_{T,j_m} is zero, or it gets decreased by $d(S - L)$.

Now we need to compare new S with the former. If we had $\mathsf{Segment}(S) \subset \mathsf{Segment}(T)$, then new $\mathsf{Segment}(S)$ is also the subset of the former $\mathsf{Segment}(T)$, so the new S is still a good set, within the range $R(i_1)$. Moreover, new $\mathsf{length}(S)$ is at most the former $\mathsf{length}(T) - 1$, as i_1 is not in $\mathsf{Segment}(S)$. Therefore the decrease of the ordered length vector in the lexicographic order is determined by the decrease in $\mathsf{length}(T)$.

Now assume $\mathsf{Segment}(S) \not\subset \mathsf{Segment}(T)$. Because $\mathsf{segment}(S) \subset \mathsf{Segment}(T)$, we must have $i_\ell = L(S)$, and thus $\mathsf{segment}(T) \subset \mathsf{Segment}(S)$. Now the same arguments that we used to compare new T with the former T apply to the comparison of new S to former S: new $\mathsf{Segment}(S)$ is contained in the former $\mathsf{Segment}(S)$, implying that new $\mathsf{Segment}(S)$ is contained in $R^{rev}(i_\ell)$, and $\mathsf{length}(S)$ is decreased by at least $d(T - L)$.

It is easy to see that only a finite number of such decreases is possible, therefore eventually we will obtain a good solution without any strong crossings. ❑

Now we can assume that no two elements of our good solution strongly cross each other and we want to characterize connected components of the graph ($\mathcal{F}$,crosses). We will ignore sets with empty $S - L$, and the remaining ones we classify as follows: $L(S)$ does not exists — S is *small*, $L(S)$ starts $\mathsf{path}(S)$ — S is *left*, $L(S)$ ends $\mathsf{path}(S)$ — S is *right*, $L(S) \in \mathsf{segment}(S)$ — S is straight.

Lemma 9 *Suppose that we have a pair of sets,* S, T *where* T *is a successor of* S, S *is not right and* T *is not left, and let* i *be the end of* $\mathsf{path}(S)$. *Then there exists no* $Q \in \mathcal{F}$ *such that* i *is in the interior of* $\mathsf{path}(Q)$.

Proof. Neither S nor T may have i in the interior of its path: otherwise S would be a right set, or T would be a left set. Suppose that this is true for some other $Q \in \mathcal{F}$. Assume w.l.o.g. that $\mathsf{path}(Q)$ starts with $L(Q)$; then the entire $\mathsf{segment}(T)$ is on $\mathsf{path}(Q)$ on which it is followed by $\mathsf{segment}(Q)$, and this implies a crossing of Q with T; because $\mathsf{path}(Q, T)$ starts with $L(Q)$, this is a strong crossing, a contradiction. ❑

Lemma 10 *Assume that no pair of successive sets in $\mathcal{F}$ satisfies the assumptions of Lemma 9. Then either all sets in $\mathcal{F}$ are left or all of them are right.*

Proof. It follows immediately from the definitions. ❑

We use Lemma 9 to break the cycle of $U - L$ into a path that ends at i; if $i \in \mathsf{segment}(T)$, then we suitably split i into two elements, i' and i'', with $d_i = d_{i'} + d_{i''}$. As a result, (U, Succ) now becomes a path, and for every crossing pair P, Q the $\mathsf{path}(P, Q)$ is (essentially) not altered.

We keep breaking the path of (U, Succ) as long as we have a successor pair satisfying Lemma 9; as a result, in a successor pair (S, T) either S is right, or T is left. A path of successor pairs may have some number of left sets, followed by a straight or small set (if any), followed by a number of right sets. Every crossing occurs within a single path.

Another observation is that a crossing between two right (left) sets is a strong crossing. Thus a right set may cross left sets only and possibly the straight set in the middle of the path. Note that if it crosses any of the left sets, it crosses all left sets that are before it. Therefore left sets that participate in crossings form the initial portion of their sub path, while the right sets that participate in crossings form the final part. As a result, the sets participating in crossings form a contiguous sub path, with the only possible exception for the separating set, if the latter exists. In turn, $L(S)$'s of the sets in a path that participate in crossings form a contiguous path P in (L, Succ_L).

Finally, if we take sets S from a contiguous sub-path of P, say P', and we take the union (or sum) or the respective $\mathsf{segment}(S)$'s, we obtain at most three contiguous intervals: that of left sets with $L(S)$ on P', that of right set, and that of the straight set.

4.3 Conclusion

We briefly sketch the idea behind the algorithm for finding a solution to MinAntLoad That is a rounding of a good solution with no strong crossing, while such a solution is within factor 1.5 from the optimum. The algorithm for MinAnt is similar.

Assume first that Lemma 9 cannot be applied. By Lemma 10 we know the existence of such linear ordering unless all the normalized sets are entirely of the left or right type and therefore no crossing exists. In the latter case we can separately break the circular order of L to get $(i_1, \ldots, i_k)$ and the circular order of $U - L$ to get $(j_1, \ldots, j_m)$ and then we can find a solution with greedy packing. Set S_a will have $L(S_a) = i_a$; $\mathsf{first\_small}(S_1) = j_1$, and given $\mathsf{first\_small}(S_a)$ we find the longest possible $\mathsf{segment}(S_a)$ such that $d(S_a) \leq 1.5$ and $S_a \subset R(i_a)$ or $S_a \subset R(\mathsf{first\_small}(S_a))$. If $a \neq k$, we define $\mathsf{first\_small}(S_{a+1}$ as the first element of $U - L$ that follows $\mathsf{segment}(S_a)$. We can run this greedy algorithm for every possible pair i_1, j_1.

Solutions that exist when Lemma 9 can be applied can be found using dynamic programming in we find all possible paths that can be obtained by successive applications of that lemma.

Suppose that we want to check if a path $(i_1, \ldots, i_k)$ may form such a path. The first case is that this path is covered with a single set; obviously we can quickly verify if this is possible. The second case is that the path can be covered with a number of right sets, followed by a straight or a small set, followed by a number of left sets (each of the three groups may be missing but there have to be at least two groups). Let $(j_1, \ldots, j_\ell)$ be the subsequence of the path that consists of elements of L, and let $(i_1, \ldots, i_k)$ represent the subsequence that consists of elements of $U - L$. We can ask two questions: can these two sequences be covered by ℓ sets and can these two sequences be covered by $\ell + 1$ sets.

The first case is that the set S_1 that is used in such a cover and such that $L(S_1) = j_1$ does not cross any other set. We form set $S_1 = \{j_1, i_1, i_2, \ldots, i_a\}$ such that $S_1 \subset R(j_1)$ and $S_1 \subset R(i_1)$ and $d(S_1) \le 1.5$ and a is maximal, and then we ask a question about covering of $(j_2, \ldots, j_k)$ and $(i_{a+1}, \ldots, i_\ell)$. The case when the set that contains j_k does not cross any other set is symmetric.

Now we consider the case in which the desired cover exists only if all right sets and left sets included in the cover participate in set crossings. There are three subcases: the central set exists and it is a straight set, it exists and it is a small set and it does not exists. We will describe the dynamic programming for the first subcase, the other two are similar and simpler.

First in every possible way we guess the central set S_c: we assume a certain $L(S_c) = j_c$ and $first\_small(S_c) = i_d$. We find the maximal $segment(S_c)$ that is consistent with our assumptions, and we abandon this assumption if $j_c \notin segment(S_c)$. Otherwise from the assumption j_c, i_d we compute the last element of $segment(S_c)$, $f_1(j_c, i_d)$.

Now we consider a prefix of $(j_1, \ldots)$, say $(j_1, \ldots, j_a)$, and possible unions of segments of sets $S_1, \ldots, S_a$ and S_c. One can see that in $(U - L, Succ_s)$ this union forms two contiguous path, one that starts at i_1 and ends at some i_b (Part 1) the other starts at i_d and ends at $f_2(j_c, i_d, j_1, j_a, i_1, i_b)$ (Part 2). We assume that $impossible$ is one of the values of f_2. We need a recursive formula for f_2 and this will be the essence of our dynamic programming solution.

The basic case is when $a = 1$; if $i_1 \neq Succ_S(i_b)$, then $f_2 = impossible$ (this means that Part 1 has to be empty). For $i_1 = Succ_S(i_b)$, we find the longest segment that we can associate with j_1 and that starts at $i = Succ_S(f_1(j_c, i_d))$, and ends at $f_1(j_1, i)$. (If S_1 is a right set, with $Segment(S_1)$ starting at i_1, then we actually have the first case.)

Now we compute $f_2(j_c, i_d, j_1, j_a, i_1, i_b)$ for $a > 1$. We consider two cases, and take the larger of the two solutions (or the only possible one). Either (i) $segment(S_a)$ ends Part 1, or (2) $segment(S_a)$ ends Part 2. For case (i), we find the smallest possible $first\_small(S_a)$ that satisfies these conditions: $segment(S_a)$ ends at i_b, $L(S_a) = j_a$; $d(S_a) \le 1.5$, $j_a \in R(first\_small(S_a))$, so it can be computed as $first\_small(S_a) = f_3(j_a, i_b)$; this implies this possibility:

$$f_2(j_c, i_d, j_1, j_a, i_1, i_b) = f_2(j_c, i_d, j_1, j_{a-1}, i_1, f_3(j_a, i_b))$$

For case (ii) we start $segment(S_a)$ at $f_2(j_c, i_d, j_1, j_{a-1}, i_1, i_b)$ and this implies this possibility:

$$f_2(j_c, i_d, j_1, j_a, i_1, i_b) = f_1(j_a, f_2(j_c, i_d, j_1, j_{a-1}, i_1, i_b))$$

Clearly, we can tabulate f_1 and f_3, so the recursive formula for f_2 can be computed in constant time, and a dynamic programming that computes f_2 fills a table with $O(n^6)$ entries.

We can use f_2 as follows: we can cover $(j_1, \dots, j_k)$ and $(i_1, \dots, i_\ell)$ with k sets if for some j_c, i_d we have

$$f_2(j_c, i_d, j_1, j_k, i_1, i_{d-1}) = i_\ell$$

One can see that in time $O(n^6)$ we can find all tuples $(j_1, j_k, i_1, i_\ell, m)$ such that $(j_1, \dots, j_k)$ together with $(i_1, \dots, i_\ell)$ can be covered with m sets from a family that forms a rounding of a good solution. This proves

Theorem 11 *There exists a polynomial time approximation algorithm for the problems of* MinAnt *and* MinAntLoad *with ratio* 3/2.

5 Acknowledgments

We thank Sudarshan Vasudevan for suggesting the problem and initial discussions. We also thank Webb Miller for supporting Jieun Jeong through NIH grant HG02238 and Martin Fürer for supporting Shiva Kasiviswanathan through NSF Grant CCR-0209099.

References

[1] L. Bao and J.J. Garcia-Luna-Aceves. Transmission scheduling in ad hoc networks with directional antennas. In *Proceedings of MOBICOM'2002*, September 2002.

[2] E. G. Coffman, M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: a survey. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard problems*, pages 46–93. PWS Publishing, Boston, 1997.

[3] A. Nasipuri, S. Ye, J. You, and R. Hiromoto. A MAC protovol for modile ad hoc networks using directional antennas. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'2000)*, 2000.

[4] A. Spyropoulos and C. S. Raghavendra. Energy efficient communication in ad hoc networks using directional antennas. In *Proceedings of IEEE INFOCOM'2002*, June 2002.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/