Karen Aardal
Bert Gerards (Eds.)

LNCS 2081

Integer Programming and Combinatorial Optimization

8th International IPCO Conference
Utrecht, The Netherlands, June 2001
Proceedings

Springer

Karen Aardal Bert Gerards (Eds.)

Integer Programming and Combinatorial Optimization

8th International IPCO Conference
Utrecht, The Netherlands, June 13-15, 2001
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editors

Karen Aardal
Universiteit Utrecht, Instituut voor Informatica en Informatiekunde
Padualaan 14, 3584 CH Utrecht, The Netherlands
E-mail: aardal@cs.uu.nl

Bert Gerards
CWI
Kruislaan 413, 1098 SJ Amsterdam, The Netherlands
E-mail: bgerards@cwi.nl

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Integer programming and combinatorial optimization : proceedings / 8th International IPCO Conference, Utrecht, The Netherlands, June 13 - 15, 2001. Karen Aardal ; Bert Gerards (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2081)
ISBN 3-540-42225-0

CR Subject Classification (1998): G.1.6, G.2.1, F.2.2

ISSN 0302-9743
ISBN 3-540-42225-0 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10839281 06/3142 5 4 3 2 1 0

Preface

This volume contains the papers selected for presentation at IPCO VIII, the Eighth Conference on Integer Programming and Combinatorial Optimization, Utrecht, The Netherlands, 2001. This meeting is a forum for researchers and practitioners working on various aspects of integer programming and combinatorial optimization. The aim is to present recent developments in theory, computation, and application of integer programming and combinatorial optimization. Topics include, but are not limited to: approximation algorithms, branch and bound algorithms, computational biology, computational complexity, computational geometry, cutting plane algorithms, diophantine equations, geometry of numbers, graph and network algorithms, integer programming, matroids and submodular functions, on-line algorithms, polyhedral combinatorics, scheduling theory and algorithms, and semidefinite programs.

IPCO was established in 1988 when the first IPCO program committee was formed. The locations and years of the seven first IPCO conferences were: IPCO I, Waterloo (Canada) 1990, IPCO II, Pittsburgh (USA) 1992, IPCO III, Erice (Italy) 1993, IPCO IV, Copenhagen (Denmark) 1995, IPCO V, Vancouver (Canada) 1996, IPCO VI, Houston (USA) 1998, IPCO VII, Graz (Austria) 1999. IPCO is held every year in which no MPS (Mathematical Programming Society) International Symposium takes place. Since the MPS meeting is triennial, IPCO conferences are held twice in every three-year period. As a rule, IPCO is held somewhere in Northern America in even years, and somewhere in Europe in odd years.

In response to the call for papers for IPCO 2001, the program committee received 108 submissions, indicating a strong and growing interest in the conference. The program committee met on January 13 and 14, 2001, in Amsterdam, The Netherlands, and selected 32 contributed papers for inclusion in the scientific program of IPCO 2001. The selection was based on originality and quality, and reflects many of the current directions in integer programming and optimization research. The overall quality of the submissions was extremely high. As a result, many excellent papers could unfortunately not be chosen.

The organizing committee for IPCO 2001 consisted of Karen Aardal, Bert Gerards, Cor Hurkens, Jan Karel Lenstra, and Leen Stougie. IPCO 2001 was organized in cooperation with the Mathematical Programming Society, and was sponsored by BETA, CQM, CWI, DONET, The Netherlands Society for OR (NGB), EIDMA, ILOG, IPA, Philips Research Labs, Technische Universiteit Eindhoven, the Technology Foundation STW, and Universiteit Utrecht.

April 2001

Karen Aardal
Bert Gerards

IPCO VIII Program Committee

Bert Gerards (Chair), CWI Amsterdam and Technische Universiteit Eindhoven
Karen Aardal, Universiteit Utrecht
Imre Barany, Rényi Intézet, Magyar Tudományos Akadémia, Budapest and University College London
Dan Bienstock, Columbia University
Michele Conforti, Universita degli studi di Padova
Michel Goemans, Massachusetts Institute of Technology
Monique Laurent, CWI Amsterdam and CNRS, France
Gerhard Woeginger, Technische Universität Graz and Universiteit Twente

Table of Contents

Two $O(\log^* k)$-Approximation Algorithms for the Asymmetric $k-$Center Problem

Aaron Archer*

Operations Research Department, Cornell University, Ithaca, NY 14853
aarcher@orie.cornell.edu

Abstract. Given a set V of n points and the distances between each pair, the k-center problem asks us to choose a subset $C \subseteq V$ of size k that minimizes the maximum over all points of the distance from C to the point. This problem is NP-hard even when the distances are symmetric and satisfy the triangle inequality, and Hochbaum and Shmoys gave a best-possible 2-approximation for this case.
We consider the version where the distances are asymmetric. Panigrahy and Vishwanathan gave an $O(\log^* n)$-approximation for this case, leading many to believe that a constant approximation factor should be possible. Their approach is purely combinatorial. We show how to use a natural linear programming relaxation to define a promising new measure of progress, and use it to obtain two different $O(\log^* k)$-approximation algorithms. There is hope of obtaining further improvement from this LP, since we do not know of an instance where it has an integrality gap worse than 3.

1 Introduction

Suppose we are given a road map of a city with n buildings where we wish to offer fire protection, along with the travel times (a distance function) between each pair of buildings, and suppose we are allowed to make k of the buildings into fire stations. Informally, the asymmetric k-center problem asks how to locate the fire stations (*centers*) in order to minimize the worst case travel time to a fire (the *covering radius*). It is common to assume the distance function is symmetric and satisfies the triangle inequality. Without the triangle inequality, it is NP-hard even to decide whether the k-center optimum is finite, by a reduction from set cover. So we require the distances to satisfy the triangle inequality, which also makes sense when we think of them as travel times. But distances may be asymmetric due to one-way streets or rush-hour traffic, so in this paper we do not require symmetry.

The asymmetric k-center problem has proven to be much more difficult to understand than its symmetric counterpart. In the early 1980's, Hochbaum and Shmoys [6,7] first gave a simple 2-approximation algorithm for the symmetric

* Supported by the Fannie and John Hertz Foundation and ONR grant AASERT N0014-97-10681.

K. Aardal, B. Gerards (Eds.): IPCO 2001, LNCS 2081, pp. 1–14, 2001.

variant. Shortly thereafter, Dyer and Frieze [4] found another 2-approximation. These results essentially closed the problem because this approximation guarantee is the best possible, by an easy reduction from set cover. However, no non-trivial approximation algorithm was known for the asymmetric version until Panigrahy and Vishwanathan [14,13] gave an $O(\log^* n)$-approximation more than ten years later.

The k-center problem is one of the basic clustering problems. Uncapacitated facility location and k-median are two other prominent ones, and both of these admit constant-factor approximations in the symmetric case (see [2] for the current best factors). But in the asymmetric case, the best results yield $O(\log n)$ factors [5,10], relying on clever applications of the greedy set cover algorithm. Moreover, a natural reduction from set cover gives inapproximability results that match these bounds up to a constant [1]. In stark contrast, it is now widely believed that a constant-factor approximation algorithm should exist for the asymmetric k-center problem. This conjecture is prompted by the $O(\log^* n)$ result of [13], especially since no hardness of approximation result is known beyond the lower bound of 2 inherited from the symmetric version.

In this paper we introduce a natural linear programming relaxation that seems a promising step towards a constant-factor approximation. Our LP is essentially a set cover LP derived from an unweighted directed graph, where the optimal k-center solution corresponds to an optimal integral cover (dominating set). Whereas this LP has an $O(\log n)$ integrality gap for the set cover problem, it behaves better in our context because k-center has a different objective function. The LP objective is the number of fractional centers chosen, whereas the k-center objective is the covering radius. We present two $O(\log^* k)$-approximation algorithms for the asymmetric k-center problem, both of which use the fractional centers given by the LP solution as a guide. There is hope that our LP might be used to obtain further improvements since we do not know of any instances where it has an integrality gap worse than 3 for the asymmetric k-center problem.[1]

The $O(\log^* n)$ algorithm of [13] is entirely combinatorial. Essentially, it uses the greedy set cover algorithm to choose some (too many) centers, and then recursively covers the centers until the correct number remain. The advantage of working with our LP is that it gives us fractional center values to guide our algorithm and measure its progress. Rather than choosing centers to cover the nodes of our graph, we instead try to efficiently cover the fractional centers.

The crux of the first algorithm is our ExpandingFront routine. We select centers greedily, hoping to cover lots of fractional centers, because we are covering within two steps what these fractional centers covered within one. For this strategy to succeed, we need to somehow guarantee that the centers we choose cover many fractional centers. The key idea here is that as we choose more centers, the set A of active centers that remain to be covered shrinks, as does the number of centers necessary to cover them. We show that the amount of progress we make at each step grows substantially as the size of the optimal

[1] Here we use integrality gap in a slightly non-standard sense. See the remarks in Section 3 for details.

fractional cover of A shrinks, so it is important to reduce this quantity rapidly. The fractional center values from the LP allow us to enforce and measure this shrinkage.

We can also use the fractional centers to modify the algorithm of [13] to obtain the same $O(\log^* k)$ performance guarantee. Our RECURSIVECOVER routine uses them to obtain an initial cover with fewer centers than the initial cover produced in [13], then employs precisely the same recursive set cover scheme to finish.

Since achieving a constant-factor approximation seems difficult, it is natural to attempt to find a bicriterion approximation algorithm that blows up both the number of centers and the covering radius by a constant factor. This might seem to be a significantly easier task, but in fact [13] shows this would essentially give us a constant-factor unicriterion approximation. This is because we can preprocess the problem with the REDUCE routine of Section 6, and postprocess our solution by recursively covering the centers we chose, as in Section 5.

2 Formal Problem Definition and Notation

The input to the asymmetric k-center problem is a parameter k, a set V of n points, and a matrix D specifying a distance function $d : V \times V \to \mathbb{R}_+ \cup \{\infty\}$. Think of $d(u,v)$ as the distance *from* u *to* v. We require our distance function to obey the triangle inequality, but not symmetry. That is, $d(u,w) \leq d(u,v) + d(v,w)$ for all $u,v,w \in V$, but $d(u,v)$ may differ from $d(v,u)$.

Any set of *centers* $C \subseteq V$ with $|C| \leq k$ is a *solution* to the k-center problem. The *covering radius* of a solution C is the minimum distance R such that every point is within R of the set C. (It could be ∞.) The goal is to find the solution with the minimum covering radius R^*.

We will find it convenient to work with unweighted directed graphs on node set V instead of on the space (V,D) directly. With respect to a directed graph $G = (V,E)$, we define (for $i \in \mathbb{Z}_+$)

$$\Gamma_i^+(u) = \{v \in V : G \text{ contains a directed path from } u \text{ to } v \text{ using at most } i \text{ edges}\}.$$

Conversely, $\Gamma_i^-(u)$ is the set of nodes *from* which u can be reached by a directed path using at most i edges. We suppress the subscript when $i = 1$. Thus $\Gamma^+(u)$ is u plus its out-neighbors, and $\Gamma^-(u)$ is u plus its in-neighbors. For $S \subseteq V$ we define $\Gamma_i^+(S)$ and $\Gamma_i^-(S)$ analogously. In G, we say S *covers* T *within* i (or i-*covers* T) if $\Gamma_i^+(S) \supseteq T$. When $i = 1$, we just say S *covers* T.

For $R \geq 0$, we define the graph $G_R = (V, E_R)$, where $E_R = \{(u,v) : d(u,v) \leq R\}$. The essential connection here is that there exist k centers that cover all of G_R if and only if $R \geq R^*$. Thus we can binary search for the optimal radius R^* and work in G_{R^*}. Finding an i-cover in this graph will yield an i-approximation in the original space, since traversing each edge in the graph corresponds to moving at most R^* in the original space. Moreover, in the worst case this analysis is tight, because our given distances could be those induced by shortest directed paths in an unweighted graph.

For $y \in \mathbb{R}^S$, let $y(S)$ denote $\sum_{v \in S} y_v$. For a function g, let $g^{(i)}$ denote the function iterated i times. Finally, define $\log^* x = \min\{i : \log^{(i)} x \leq \frac{3}{2}\}$.

3 Overview of the Algorithm

We describe a polynomial time relaxed decision procedure AKC (see Figure 2), which takes as input an asymmetric k-center instance and a guess at the optimal radius R, and either outputs a solution of value $O(R \log^* k)$ or correctly reports that $R < R^*$. Since $R^* = d(u, v)$ for some $u, v \in V$, there are only $O(n^2)$ different possible values for R^*. We binary search on R, using at most $O(\log n)$ calls to AKC to yield a solution of value $O(R \log^* k)$ for some $R \leq R^*$, which gives our $O(\log^* k)$-approximation algorithm. Thus, by guessing the covering radius R, we immediately convert the optimization problem into a promise problem on G_R and thereafter think only in terms of this graph and others derived from it.

Theorem 1 *Given a parameter R and an asymmetric k-center instance (V, D, k) whose optimum is R^*, AKC (V, D, k, R) runs in polynomial time and either proves that $R < R^*$ or outputs a set C of at most k centers covering V within $R(3 \log^* k + O(1))$.*

The general framework we use was introduced by [13]. There are two substantive components – a preprocessing phase called REDUCE, and the heart of the algorithm, which we call AUGMENT. In addition, our version of AKC solves an LP. We can solve the LP in polynomial time, and it will be clear that our algorithms for REDUCE and AUGMENT run in polynomial time. Thus AKC runs in polynomial time.

We present two different ways to implement the AUGMENT phase, EXPANDINGFRONT (Section 4) and RECURSIVECOVER (Section 5), each of which improves the approximation guarantee from $O(\log^* n)$ to $O(\log^* k)$. For completeness, we also describe the REDUCE phase in Section 6, slightly sharpening the analysis given in [13] to improve the constant inside the $O(\log^* k)$ from 5 to 3. We now describe how REDUCE and AUGMENT fit together.

Assume we have a graph G, and we are promised that there exist k centers covering G. Then AUGMENT finds at most $\frac{3}{2}k$ centers that cover all of G within $\log^* k + O(1)$. How do we obtain a solution that uses only k centers? Roughly speaking, we will prove that if there are k centers that cover V, then there exist $\frac{2}{3}k$ centers 3-covering V. So we can run AUGMENT in G^3, the cube of G, to find k centers that cover all of V in G within $3 \log^* k + O(1)$.

More precisely, the REDUCE phase preprocesses the graph G by choosing an initial set C of at most k centers consisting of some special nodes, called *center capturing vertices*. These centers already cover part of V within some constant radius. The set of nodes that remain to be covered we call the *active set*, and denote it by A. We prove that there exist $p \leq \frac{2}{3}(k - |C|)$ centers 3-covering A. Then, again roughly speaking, we use the AUGMENT procedure in G^3 to augment C by at most $\frac{3}{2}p$ new centers to a set of at most k centers covering V within $\log^* p + O(1)$ in G^3.

Our LP, with respect to $G = (V, E)$ and $A \subseteq V$

$$\begin{aligned} &\min\ y(V) \\ &\text{s.t. } y(\Gamma^-(v)) \geq 1 \text{ for all } v \in A \\ &\quad y \geq 0 \end{aligned}$$

Fig. 1. Here is our set cover-like LP, defined with respect to some graph G with nodes V and an active set A of nodes to be covered. Recall the notation $y(S) = \sum_{v \in S} y_v$.

Both EXPANDINGFRONT and RECURSIVECOVER use the linear program of Figure 1. In this LP, if we further restrict $y \in \{0,1\}^V$, the resulting integer program asks for the smallest set of centers necessary to cover A. This is just a set cover problem where A is the set of elements to be covered, and $\{A \cap \Gamma^+(v) : v \in V\}$ is the collection of sets. We think of any solution y as identifying fractional centers, so y is a *fractional cover* of A. We note that our AUGMENT phase does not require there to exist an integral cover with p centers, nor does it require y to be an optimal fractional cover. It suffices to have any fractional cover with at most p fractional centers. We use these fractional centers to guide the AUGMENT procedure.

The following two theorems (proved in Sections 6 and 4.2, respectively) summarize the technical details. When put together with the precise description of AKC in Figure 2, Theorem 1 follows.

Theorem 2 *Given a directed graph G for which there is a cover using k centers,* REDUCE (G) *outputs a set of centers C and an active set $A = V \backslash \Gamma_4^+(C)$ such that there exists a 3-cover of A using at most $\frac{2}{3}(k - |C|)$ centers.*

Theorem 3 *Suppose $A = V \backslash \Gamma^+(C)$ in G, y is a fractional cover of A, and $p = y(V)$. When* AUGMENT (G, A, C, y, p) *is implemented with either* EXPANDINGFRONT *or* RECURSIVECOVER, *it augments C by at most $\frac{3}{2}p$ additional centers to cover A within $\log^* p + O(1)$ in G.*

We show two different ways to implement the AUGMENT phase, each producing a $(\log^* p + O(1))$-cover. The first, EXPANDINGFRONT, introduces the idea of choosing centers to cover fractional centers, and shows how to use the LP to make our future choices more efficient by reducing the number of fractional centers necessary to cover the new active set. The second, RECURSIVECOVER, shows how to combine the fractional center covering idea with the recursive set cover technique of [13] to obtain the same improvement.

Remarks. Our main contribution is in using the LP solution to define a new notion of progress based on covering fractional centers. We believe that this LP and the techniques introduced here may lead to a constant-factor approximation. Even though our LP has a $\Theta(\log n)$ integrality gap for the set cover problem,

AKC (V, D, k, R)

$(C, A) \leftarrow$ REDUCE (G_R)
$p \leftarrow \frac{2}{3}(k - |C|)$
$\hat{G} \leftarrow G_{3R}$ plus the edges $\{(u, v) : u \in C, v \in \Gamma_4^+(C)\}$
(where $\Gamma_4^+(C)$ is interpreted in G_R)
solve the linear program of Figure 1 defined by $\hat{G}$ and A to get y
if $y(V) > p$ then STOP and conclude $R < R^*$
else
    $p \leftarrow y(V)$
    $C \leftarrow$ AUGMENT $(\hat{G}, A, C, y, p)$
    output C

Fig. 2. Formal description of AKC.

we do not know of any examples where its integrality gap for the asymmetric k-center problem is worse than 3. That is, we are not aware of any graphs G for which there is a fractional cover using k fractional centers but there is no integral 3-cover using $\lceil k \rceil$ centers.

There is a graph G on 66 nodes that can be covered with 6 fractional centers, while the smallest integral 2-cover uses 7 centers [12]. This is the smallest example we know that establishes the LP integrality gap of 3. Probabilistic constructions yield an infinite family of graphs on n nodes having a fractional cover using k fractional centers but admitting no integral 2-cover using fewer than $\Omega(k\sqrt{\log n})$ centers.

4 Augment Phase: ExpandingFront

Recall that the AUGMENT phase takes as input a directed graph G, a set of already chosen centers C, an active set $A = V \backslash \Gamma^+(C)$ of nodes not already covered by C, and a solution y to the LP of Figure 1, with $p = y(V)$. We wish to select an additional $\frac{3}{2}p$ centers that, along with C, cover A within $\log^* p + O(1)$ in G.

4.1 Motivation and Description of ExpandingFront

For motivation, consider the case where $C = \emptyset$ so A is all of V. If there exists a cover using only q integral centers, then clearly one of these centers v must cover at least $\frac{p}{q}$ units of fractional centers. That is, $y(\Gamma^+(v)) \geq \frac{p}{q}$. It turns out that this result holds also when there are q fractional centers covering p fractional centers (see Lemma 4 below, with $z = y$), and we can apply this observation with $p = q$. Thus, there exists a node v covering a full unit of fractional centers.

It makes sense to choose such a node as a center, because it 2-covers what these fractional centers covered. If we could manage to continue covering one

new unit of fractional centers with each new center we select, then we would use only p centers to cover all the fractional centers, which means we would 2-cover all of V. This would yield a 2-approximation.

The problem is that when we choose a new center v, we remove $\Gamma^+(v)$ from A, the active set of fractional centers yet to be covered. To see how many units of fractional centers our next greedily chosen center covers, we use Lemma 4. Since $y(A)$, the amount of active fractional centers, decreases, while $y(V)$ is still p, the best guarantee we can make is that each new greedily-chosen center covers at least a $\frac{1}{p}$ fraction of the remaining active fractional centers.

Lemma 4 *Let $G = (V, E)$ be a directed graph, $A \subseteq V$, and $z \in \mathbb{R}_+^A$ be any set of non-negative weights on A. If $y \in \mathbb{R}_+^V$ is a fractional cover of A, that is, y is any feasible solution to the linear program of Figure 1, then there exists $v \in V$ such that*

$$z(\Gamma^+(v) \cap A) \geq \frac{z(A)}{y(V)}. \tag{1}$$

Proof. We take a weighted average of $z(\Gamma^+(v) \cap A)$ over $v \in V$.

$$\begin{aligned}
\frac{1}{y(V)} \sum_{v \in V} y_v z(\Gamma^+(v) \cap A) &= \frac{1}{y(V)} \sum_{v \in V} \sum_{u \in \Gamma^+(v) \cap A} y_v z_u \\
&= \frac{1}{y(V)} \sum_{u \in A} z_u \sum_{v \in \Gamma^-(u)} y_v \\
&\geq \frac{1}{y(V)} \sum_{u \in A} z_u
\end{aligned}$$

The inequality follows because $z \geq 0$ and $y(\Gamma^-(u)) \geq 1$ for all $u \in A$. Since some term is at least as large as the weighted average, we know $z(\Gamma^+(v) \cap A) \geq \frac{z(A)}{y(V)}$ for some $v \in V$. □

We might be saved if we somehow knew that the present active set A had a fractional cover y_A with fewer than p fractional centers. We could then use y_A instead of y, so the denominator $y_A(V)$ of (1) would decrease along with the numerator $y(A)$. We can indeed reduce the denominator by the following observation: for any set S, the nodes in $\Gamma^+(S)$ do not help cover any of the nodes in $V \backslash \Gamma_2^+(S)$. To aid this discussion, we introduce some new notation.

With respect to a graph G and a current set of centers $C \subseteq V$, define $V_i = \Gamma_i^+(C) \backslash \Gamma_{i-1}^+(C)$, the nodes exactly i steps from from C in G, and $V_{\geq i} = V \backslash \Gamma_{i-1}^+(C)$, the nodes at least i steps from C in G. If we consider a directed breadth first search tree from the set C, then $V_{\geq i}$ consists of the nodes at and beyond the i^{th} level, so this set shrinks as we add centers to C.

In our motivational example (where initially $A = V$), when selecting our first center v with $y(\Gamma^+(v)) \geq 1$, let us set $A \leftarrow V_{\geq 3}$ (instead of $A \leftarrow V_{\geq 2}$). See Figure 3. Since V_1 does not help cover A, projecting y onto $V_{\geq 2}$ yields a fractional

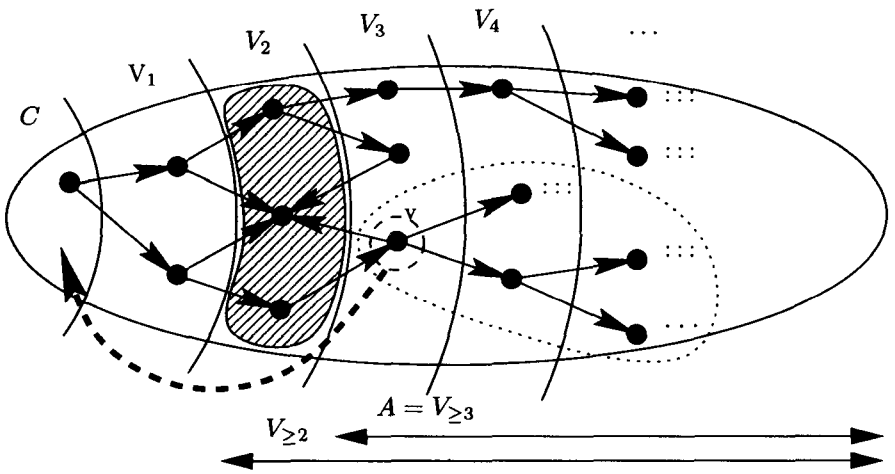

Fig. 3. This illustrates one step of the $i = 1$ phase of EXPANDINGFRONT.

cover of A, so we can replace $y(V)$ by $y(V_{\geq 2}) \leq p - 1$ in the denominator of (1) for the next greedy choice. Unfortunately, the numerator decreases to $y(A) = y(V_{\geq 3})$, which is smaller by exactly $y(V_2)$, so the next greedily chosen center may cover less than one additional unit of fractional centers.

This problem seems to be unavoidable. The difficulty is that $y(V_2)$ might be quite large, say $\frac{p}{2}$. Since these fractional centers are no longer in the active set, our chosen centers may never cover them, so they will always contribute to the denominator in (1).

The idea of EXPANDINGFRONT is to greedily choose a few centers, then "flush" the irritating fractional centers trapped at the "front" V_2, by setting $A \leftarrow V_{\geq 4}$. We can then ignore the centers in V_2 since they no longer help cover A. This expands the front (the location of the irritating nodes) to radius 3. In the bad case where $y(V_2)$ is large, we flush a lot of fractional centers. We then repeat this process. The penalty is that each time we flush, we expand the radius and so we do not get a constant-factor approximation. As it turns out, we need flush only $\log^* p + O(1)$ times. See Figure 4 for a precise description. For simplicity, the algorithm begins with phase $i = 0$, whereas this motivating discussion corresponds to the $i = 1$ phase.

Figure 3 shows part of the breadth first search tree from C at one of the steps of the $i = 1$ phase. Suppose v (circled) is the center chosen greedily from $V_{\geq 2}$. Adding it to C "pulls" it's out-tree (encircled by the dotted oval) to the left in the diagram. The shaded area is the "front," i.e. the irritating strip of nodes that cause our lower bound on $y(\Gamma^+(v) \cap A)$ from (1) to be less than one. By the end of phase 0, we had chosen $\frac{3}{4}p$ centers, reducing $y(A)$ to $\exp(-\frac{3}{4}) \cdot p \approx 0.472p$. We then moved the front one strip to the right, so we know at most $0.472p$ fractional centers are necessary to cover the current A throughout phase 1.

ExpandingFront (G, A, C, y, p)

for $i = 0, 1, 2, \ldots$ do ("phase" i)
  for $j = 1, 2, \ldots \lceil \frac{3}{4} \frac{p}{2^i} \rceil$ do (j is just a counter)
    if $y(A) < 1$ then STOP, output C
    else
      find $v \in V_{\geq i+1}$ that maximizes $y(\Gamma^+(v) \cap A)$
      $C \leftarrow C + v$
      $A \leftarrow A \backslash \Gamma^+_{i+1}(v)$ (equivalently, $A \leftarrow V_{\geq i+2}$)
  $A \leftarrow V_{\geq i+3}$ (expand the front)
  (now y_A is y projected onto $V_{\geq i+2}$)

Fig. 4. Formal description of EXPANDINGFRONT.

4.2 Analysis of ExpandingFront

At the beginning of each inner loop we have $A = V_{\geq i+2}$, so all inactive nodes are $(i+1)$-covered by C. If $y(A) < 1$, then since every $v \in A$ is covered by one unit of fractional centers, some node in $\Gamma^+_{i+1}(C)$ must cover v. Hence C $(i+2)$-covers all of V. So to prove Theorem 3 for EXPANDINGFRONT it suffices to prove Lemma 6 below, and to show that we do not choose too many centers (Lemma 10).

Definition 5 *Define the* tower function $T(n)$ *by* $T(0) = 1$ *and* $T(n+1) = e^{T(n)}$ *for* $n \geq 0$.

Lemma 6 EXPANDINGFRONT *terminates after at most* $\log^* p + O(1)$ *phases, where we call each outer loop a phase.*

Proof. Claims 7, 8 and 9 below establish that at the beginning of phase i we have $y(A) \leq \frac{p}{a_i}$ where a_i grows like a tower function. This establishes Lemma 6, since we stop once $a_i > p$. □

In phase i, since $A = V_{\geq i+2}$, we take our fractional cover y_A to be the restriction of y to the nodes $V_{\geq i+1}$, and we work on decreasing $y(A)$, which we can upper bound using Claim 7 below. Then by expanding the front at the end of the phase, our old $y(A)$ becomes our new $y_A(V)$.

Claim 7 *If at the beginning of a phase we have* $y_A(V) = a$ *and we choose* b *centers in the phase, then we reduce* $y(A)$ *at least by a factor of* $e^{-\frac{b}{a}}$.

Proof. Since y_A covers A, each chosen center v satisfies $y(\Gamma^+(v) \cap A) \geq \frac{y(A)}{a}$ by Lemma 4. In phase i, choosing v as a center actually reduces $y(A)$ by $y(\Gamma^+_{i+1}(v) \cap A)$, but we know how to account only for the reduction due to $y(\Gamma^+(v) \cap A)$. Thus, each of the b new centers reduces $y(A)$ by a factor of $(1 - \frac{1}{a}) \leq \exp(-\frac{1}{a})$. □