One for the Price of Two: A Unified Approach for Approximating Covering Problems *

Reuven Bar-Yehuda
Computer Science Department, Technion - IIT, Haifa 32000 Israel
e-mail: reuven@cs.technion.ac.il
http://www.cs.technion.ac.il/~reuven

August 3, 1998

Abstract

We present a simple and unified approach for developing and analyzing approximation algorithms for covering problems. We illustrate this on approximation algorithms for the following problems: Vertex Cover, Set Cover, Feedback Vertex Set, Generalized Steiner Forest and related problems.

The main idea can be phrased as follows: iteratively, pay two dollars (at most) to reduce the total optimum by one dollar (at least), so the rate of payment is no more than twice the rate of the optimum reduction. This implies a total payment (i.e., approximation cost) $\leq$ twice the optimum cost.

Our main contribution is based on a formal definition for covering problems, which includes all the above fundamental problems and others. We further extend the Bafna, Berman and Fujito extension of the Local-Ratio theorem. Our extension eventually yields a short generic r-approximation algorithm which can generate most known approximation algorithms for most covering problems.

Another extension of the Local-Ratio theorem to randomized algorithms gives a simple proof of Pitt's randomized approximation for Vertex Cover. Using this approach, we develop a modified greedy algorithm, which for Vertex Cover, gives an expected performance ratio ≤ 2.

Keywords: Approximation Algorithm, Local Ratio, Primal-Dual, Covering Problems, Vertex Cover, Set Cover, Feedback Vertex Set, Generalized Steiner Forest, Randomized Approximations.

1 Introduction

Most algorithms for covering problems can be viewed according to two different approaches: the primal-dual approach, and the local-ratio approach. The common characteristic that we find in algorithms for covering problems is the use of weight reductions (or deletion, which is a special case). We are interested in viewing algorithms from the viewpoint of the Local-Ratio theorem [4], and its extensions by Bafna, Berman and Fujito [2].

Following the work of Bafna et al., several studies have shown unified approaches for various families of covering problems. Fujito [15] gave a generic algorithm for node deletion. Chudack,

*This research was supported by the fund for the promotion of research at the Technion.

Goemans, Hochbaum and Williamson [11] united algorithms for the Feedback Vertex Set problem presented by Bafna et al., [2] and by Becker and Geiger [9], using the primal-dual principle. Agrawal, Klein and Ravi [1] dealt with the Generalized Steiner Forest problem. Following their work, several generalizations/unifications have recently been done, by Williamson, Goemans, Mihail, and Vazirani [28], by Goemans and Williamson [17] and by Bertsimas and Teo [10]. Bertsimas et al. present a unified view of the primal-dual approach for covering problems.

In this work, we try to combine all of these results into a unified definition for covering problems, and present a new local-ratio principle, which explains, with relative ease, the correctness of all the above algorithms. Like the others, we use reductions of the weights in the graph, but we view this in the following way:
We try to measure the "effectiveness" of the reduction we are making; as a result of the reduction, there may be a change in the optimum. The reduction incurs a cost, and we would like the payment to be bounded by a constant r times the change in the problem's optimum.

This principle of "effectiveness" can also be viewed in terms of expected values, when the reduction we make is chosen according to some random distribution. This yields an extension of the local-ratio principle in another direction; in this extension, we are interested in bounding the expected change in the optimum as related to the expected cost.

This approach gives us a simple proof of Pitt's randomized approximation for Vertex Cover.

In addition, we get a new and interesting result: while a deterministic greedy algorithm [12] does not have a constant performance ratio, even for Vertex Cover, a randomized greedy algorithm has a performance ratio of 2.

1.1 LP Formulations Overview

As mentioned earlier, algorithms for covering problems can be analyzed according to the primal-dual approach (see, e.g., recent surveys [18, 19, 25] and [27, Chapter 13]). The primal-dual schema uses the following formulation for covering problems:

$$\begin{array}{lll} \text{(IP)} & \min \sum_{j=1}^{n} w_j x_j & \\ & \text{s.t.} \ \sum a_{ij} x_j \geq c_i, & \forall i \in \{1, \ldots, m\} \\ & \quad x_j \geq 0 & \\ & \quad x_j \in \{0,1\} & \end{array}$$

where all parameters a_{ij}, w_j are nonnegative.

An r-approximation primal-dual algorithm starts with an infeasible primal solution and a feasible dual solution, and iteratively improves the feasibility of the primal solution and the optimality of the dual solution, while keeping the primal solution integral. The improvements are made so that the algorithm will terminate with a feasible primal solution and a feasible dual solution, which satisfy the primal complementary slackness conditions, and the *relaxed* dual complementary slackness conditions (i.e., for $i \in \{1, \ldots, m\}$ $y_i = 0$ or $\sum_{j=1}^{n} a_{ij} x_j \leq r b_i$).

When using duality to design an approximation algorithm for a minimizing integer program, the lowest approximation ratio one can hope to achieve is the maximum ratio of an optimal integral solution and an optimal fractional solution. This ratio is called the *integrality gap*.

Another method for designing approximation algorithms which is limited by the integrality gap is called *Rounding* (See, e.g., [19] and [27, Chapter 11]). In order to get a good approximation, we can solve the LP-relaxation of the original integer program and then convert it into an integral solution, without increasing the cost of the solution by much. In some cases the algorithm uses the *Half Integrality* of the optimal solution to achieve a 2-approximations

(See, e.g., [20, 21, 22]). Note that for some of these problems there exist faster and simpler local-ratio approximation algorithms (for recent applications see [5, 6, 8]).

In [10] Bertsimas and Teo propose a generic primal-dual framework for integer programming problems of the covering type that uses valid inequalities in its design. Bertsimas et al. define a relationship between the set of valid inequalities and the set of minimal solutions to the covering problems called *strength*, which is the primal-dual counterpart to the local-ratio concept of *effectiveness*. They describe a primal-dual algorithm as follows: at each step of the algorithm a valid inequality is chosen with strength as small as possible, this inequality is then used to decide which one of the primal variables will be fixed to 1. After a feasible primal solution is found, the reverse deletion step ensures that the solution obtained is a minimal solution.

We believe that the local-ratio approach is more natural for covering problems, because it uses weight reductions and does not require knowledge in linear programming. We think that it is combinatorially easier to construct an r-effective weight function than a valid inequality of strength r. This is especially true when dealing with problems for which we need global weight reductions (e.g., Feedback Vertex Set problem [2] and the Generalized Steiner Forest problem [17]) and when analyzing approximation algorithms which use several types of weight reductions as opposed to several types of inequalities (e.g., [4]).

1.2 Definitions

We are interested in defining a covering problem in such a way that it will contain the following problems: Vertex Cover, Set Cover, Feedback Vertex Set, Generalized Steiner Forest, Min 2-Sat, Min Cut, etc.

For a general covering problem, the input is a triple $(X, f : 2^X \to \{0,1\}, \omega : X \to R^+)$, where X is a finite set, f is monotone, i.e., $A \subseteq B \Rightarrow f(A) \leq f(B)$, and $f(X) = 1$. For a set $C \subseteq X$, we define $\omega(C) = \sum_{x \in C} \omega(x)$, which we call the *weight* of C. We say that $C \subseteq X$ is a *cover* if $f(C) = 1$.

The objective of the problem is to find a cover $C^* \subseteq X$ satisfying

$$\omega(C^*) = \min\{\omega(C) : C \subseteq X \text{ and } f(C) = 1\}.$$

Such a cover is called an *optimum cover*.

A cover $C \subseteq X$ is called an *r-approximation* if $\omega(C) \leq r \cdot \omega(C^*)$, where C^* is an optimum cover. An algorithm A is called an *r-approximation* for a family of covering problems if, for every input (X, f, ω) in the family, A returns a cover which is an r-approximation.

1.3 Overview

This paper describes a unified approach for achieving r-approximations for covering problems. In Section 2 we describe the general idea of the technique, formalized by the Local-Ratio theorem. In Section 3 we show some applications of this technique to developing algorithms for the Vertex Cover problem. In Section 4 we apply the technique to Set Covering. In Section 5 we present an extension for dealing with randomized approximations. In Section 6 we describe an enhancement of the technique which allows us to deal with some recent approximations. In Section 7 we use the enhancement to describe and prove a 2-approximation algorithm for the Feedback Vertex Set problem, in Section 8 we use it for a 2-approximation for the Generalized Steiner Forest problem, and in Section 9 we present a generic r-approximation algorithm which can generate all deterministic algorithms presented in this paper.

2 First Technique: Pay for Every Weight Reduction

Given a triple (X, f, ω), our method for achieving an approximation cover is by manipulating the weight function ω. Denote the optimum cover of (X, f, ω) by $\mathrm{OPT}(\omega)$. The following is a fundamental observation for our technique:

The Decomposition Observation: For every two weight functions ω_1, ω_2,

$$\mathrm{OPT}(\omega_1) + \mathrm{OPT}(\omega_2) \leq \mathrm{OPT}(\omega_1 + \omega_2)$$

We select some function $\delta : X \to R^+$, s.t. $\forall_{x \in X}\ 0 \leq \delta(x) \leq \omega(x)$

The function δ is used to reduce the weights, giving a new instance $(X, f, \omega - \delta)$. The value of $\mathrm{OPT}(\omega - \delta)$ may be smaller than $\mathrm{OPT}(\omega)$, and we denote their difference by $\Delta\mathrm{OPT} = \mathrm{OPT}(\omega) - \mathrm{OPT}(\omega - \delta)$. By the Decomposition Observation, $\Delta\mathrm{OPT} \geq \mathrm{OPT}(\delta)$.

This means that by reducing the weights by δ, we reduce the total optimum by at least $\mathrm{OPT}(\delta)$. How much are we willing to pay for this reduction in the optimum? Since our final objective is an r-approximation, we would like to pay no more than $r \cdot \mathrm{OPT}(\delta)$.

Let us be generous and pay for all weight reductions we make, i.e., define $\Delta\mathrm{COST}$ to be $\Delta\mathrm{COST} = \sum_{x \in X} \delta(x) = \delta(X)$. Therefore, what we need in order to get an r-approximation is

$$\delta(X) \leq r \cdot \mathrm{OPT}(\delta)$$

or, in the terminology of the title: we are willing to pay no more than r times the gain in optimum reduction.

Let us now formalize the technique. Let (X, f, ω) be a covering problem triple. The function δ is called r-effective if:

1. $\forall_{x \in X}\ 0 \leq \delta(x) \leq \omega(x)$
2. $\delta(X) \leq r \cdot \mathrm{OPT}(\delta)$.

Now, consider the following general algorithm:

Algorithm $A(X, f, \omega)$

Select r-effective δ
Call algorithm $B(X, f, \omega - \delta)$ to find a cover C
Return C

Theorem 1*:* *The Local-Ratio Theorem:*
If B returns a cover C s.t. $(\omega - \delta)(C) \leq r \cdot OPT(\omega - \delta)$
then $\omega(C) \leq r \cdot OPT(\omega)$

Proof: $\omega(C) =$

$= (\omega - \delta)(C) + \delta(C)$	[linearity of ω and δ]
$\leq (\omega - \delta)(C) + \delta(X)$	[$\delta(C) \leq \delta(X)$]
$\leq r \cdot \mathrm{OPT}(\omega - \delta) + r \cdot \mathrm{OPT}(\delta)$	[B's property and the r-effectiveness of δ]
$\leq r \cdot \mathrm{OPT}(\omega)$	[Decomposition Observation]

□

This is illustrated in the next section by a sequence of 2-approximation algorithms for the Vertex Cover problem.

3 The Vertex Cover Problem

Let $G = (V, E)$ be a simple graph with vertices V and edges $E \subseteq V^2$. The *degree* of a vertex $v \in V$ is defined by $d(v) = |\{e \in E : v \in e\}|$. The *maximum vertex degree* is defined by $\Delta_v = \max\{d(v) : v \in V\}$. A set $C \subseteq V$ is called a *vertex cover* of $G = (V, E)$ if every edge has at least one endpoint in C, i.e., $\forall_{e \in E}\ e \cap C \neq \phi$. The Vertex Cover problem (VC) is: given a simple graph $G = (V, E)$ and a weight $\omega(v) \geq 0$ for each $v \in V$, find a vertex cover C with minimum total weight. The Vertex Cover problem (VC) is NP-hard even for planar cubic graphs with unit weight [16]. For unit-weight Vertex Cover, Gavril (see [16]) suggested a linear time 2-approximation algorithm. For the general Vertex Cover problem, Nemauser and Trotter [24] developed a local optima algorithm that (trivially) implies a 2-approximation.

Until today, the best ratio known is 2. The first linear time algorithm was found by Bar-Yehuda and Even [3]. Their proof uses the primal-dual approach. It took a few more years to find a different kind of proof – the local-ratio theorem [4].

3.1 Bar-Yehuda & Even's 2-Approximation Algorithm

The main idea is as follows: let $\{x, y\} \in E$, and let $\epsilon = \min(\omega(x), \omega(y))$. Reducing $\omega(x)$ and $\omega(y)$ by ϵ has $\Delta\text{COST} = 2\epsilon$ while $\Delta\text{OPT} \geq \epsilon$, since one of the two vertices x, y participates in any optimum. So we pay 2ϵ in order to reduce the total optimum by at least ϵ, repeating this process until the optimum reaches zero. The total cost will be no more than twice the optimum.

Therefore, the 2-effective δ selected is

$$\delta(v) = \begin{cases} \epsilon & v \in \{x, y\} \\ 0 & \text{else} \end{cases}$$

Algorithm BarEven$(G = (V, E), \omega)$

For each $e \in E$
  Let $\epsilon = \min\{\omega(v) | v \in e\}$
  For each $v \in e$
    $\omega(v) = \omega(v) - \epsilon$
Return $C = \{v : \omega(v) = 0\}$

Theorem 2: *Algorithm BarEven is a 2-approximation.*

Proof: By induction on $|E|$.

The first step uses δ, which is 2-effective, since $\text{OPT}(\delta) = \epsilon$ and $\delta(V) = 2\epsilon$. The remaining steps perform a 2-approximation for $(G, \omega - \delta)$, by the induction hypothesis, and the Local Ratio Theorem. □

For primal-dual proof, see [3]. For first local-ratio proof, see [4].

3.2 Clarkson's Greedy Modification

Clarkson [13] uses the primal-dual approach to get the following 2-approximation algorithm.

Algorithm Clarkson $(G = (V, E), \omega)$

$C = \phi$
While $E \neq \phi$
  Let $x \in V$ with minimum $\epsilon = \frac{\omega(x)}{d(x)}$
  For each $y \in \Gamma(x)$ (the neighbors of x)
    $\omega(y) = \omega(y) - \epsilon$
  Add x to C and remove it from G
Return C

Theorem 3: *Clarkson's Algorithm is a 2-approximation.*

Proof: The δ that the algorithm chooses is $\delta(x) = \omega(x)$, $\delta(y) = \frac{\omega(x)}{d(x)}$ for every $y \in \Gamma(x)$, and 0 for all others. We need to show that δ is 2-effective, i.e. (1) $\forall_{v \in V}\ \delta(v) \leq \omega(v)$ (2) $\delta(V) \leq 2 \cdot \mathrm{OPT}(\delta)$.

1. for $v = x$ we defined $\delta(x) = \omega(x)$. For $v \in \Gamma(x)$, we defined $\delta(v) = \frac{\omega(x)}{d(x)} \leq \frac{\omega(v)}{d(v)} \leq \omega(v)$. For all others, $\delta(v) = 0 \leq \omega(v)$.

2. $\delta(V) = \delta(x) + \delta(\Gamma(x)) + \delta(V - \{x\} - \Gamma(x)) = \omega(x) + \omega(x) + 0 = 2 \cdot \omega(x) = 2 \cdot \mathrm{OPT}(\delta)$

The rest of the proof is by induction, using the Local Ratio Theorem. □

3.3 Subgraph Removal

Suppose we are given a family of graphs for which we wish to develop a 1.5-approximation. Getting rid of triangles is "free". To get rid of a triangle, say x, y, z, with weights $\epsilon = \omega(x) \leq \omega(y) \leq \omega(z)$, define $\delta(x) = \delta(y) = \delta(z) = \epsilon$, and $\forall_{v \notin \{x,y,z\}}\ \delta(v) = 0$. Obviously, $\mathrm{OPT}(\delta) = 2\epsilon$, while $\delta(\{x, y, z\}) = 3\epsilon$, and therefore δ is 1.5-effective.

This was helpful, for example, in developing an efficient 1.5-approximation for planar graphs, see [4].

The $(2 - \log\log n/(2\log n))$-approximation algorithm [4] uses this approach to "get rid" of odd cycles of length $\leq 2k - 1$, where $k \approx 2\log n/\log\log n$. Let $C \subseteq V$ be a cycle of length $2k - 1$. Generalizing the idea for triangles, let $\epsilon = \min\{\omega(v) : v \in C\}$, and define $\delta(v) = \epsilon$ if $v \in C$, and 0 otherwise. In this case, $\delta(C)/\mathrm{OPT}(\delta) \leq (2k - 1)/k = 2 - \frac{1}{k}$.

Bar-Yehuda and Even's linear time algorithm for VC can also be viewed as a subgraph removal algorithm, where the subgraph removed each time is an edge.

4 Vertex Cover in Hypergraphs

Let $G = (V, E)$ be a simple[1] hypergraph with vertices V and edges $E \subseteq 2^V$. The *degree of a vertex* $v \in V$ is defined by $d(v) = |\{e \in E : v \in e\}|$. The *maximum vertex degree* is defined by $\Delta_v = \max\{d(v) : v \in V\}$. The *degree of an edge* $e \in E$ is defined by $d(e) = |e|$. The *maximum edge degree* is defined by $\Delta_e = \max\{|e| : e \in E\}$. A set $C \subseteq V$ is called a *vertex cover* of $G = (V, E)$ if every edge $e \in E$ has at least one endpoint in C (i.e., $e \cap C \neq \phi$). The Hypergraph Vertex Cover problem (HVC) is: given $G = (V, E)$, and for each $v \in V$ a weight $\omega(v) \geq 0$, find a cover with minimum total weight.

[1]A hypergraph is called simple if no edge is a subset of another edge.

This generalization of the VC problem is exactly the Set Covering problem (SC).

Algorithm BarEven is written in hypergraph terminology, and therefore approximates HVC as well.

Theorem 4: *Algorithm BarEven is a Δ_e-approximation for HVC.*

Proof: For $e \in E, \epsilon = \min\{\omega(v) : v \in e\}$. Define $\delta(v) = \epsilon$ for all $v \in e$ and 0 otherwise.

Since $\mathrm{OPT}(\delta) = \epsilon$ and $\delta(V) = \delta(e) = |e| \cdot \epsilon \leq \Delta_e \cdot \epsilon$, this implies that δ is Δ_e-effective. □

5 Randomized r-approximations

Let us select $\delta : X \to R^+$ from some random distribution. Such a random distribution is called r-effective if

1. $\forall_{x \in X}\ 0 \leq \delta(X) \leq \omega(x)$ and
2. $Exp[\delta(X)] \leq r \cdot Exp[\mathrm{OPT}(\delta)]$

Intuitively: the randomized rate of payment is no more than r times the randomized rate of the optimum reduction.

Now consider the following general algorithm:

Algorithm $A(X, f, \omega)$

Select $\delta : X \to R^+$ from some r-effective random distribution
Call algorithm $B(X, f, \omega - \delta)$ to find a cover C
Return C

Theorem 5: ***The Randomized Local-Ratio Theorem:***
If B returns a cover C s.t. $Exp[(\omega - \delta)(C)] \leq r \cdot Exp[OPT(\omega - \delta)]$,
then $Exp[\omega(C)] \leq r \cdot OPT(\omega)$.

Proof: $Exp[\omega(C)] =$

$$
\begin{aligned}
&= Exp[(\omega - \delta)(C)] + Exp[\delta(C)] \\
&\leq r \cdot Exp[\mathrm{OPT}(\omega - \delta)] + r \cdot Exp[\mathrm{OPT}(\delta)] && \text{[from the properties of } B \text{ and of the distribution]} \\
&= r \cdot Exp[\mathrm{OPT}(\omega - \delta) + \mathrm{OPT}(\delta)] \\
&\leq r \cdot Exp[\mathrm{OPT}(\omega)] && \text{[by the Decomposition Observation.]} \\
&= r \cdot \mathrm{OPT}(\omega).
\end{aligned}
$$

□

An algorithm A that produces a cover C satisfying $Exp[\omega(C)] \leq r \cdot \mathrm{OPT}(\omega)$ is called a *randomized-r-approximation.*

5.1 Pitt's Randomized-2-Approximation Algorithm

Algorithm Pitt $(G = (V, E), \omega)$

$C = \{x : \omega(x) = 0\}$ and delete C from G
While $E \neq \phi$
  Select $e \in E$
  Let $\bar{\omega} = \frac{1}{\sum_{x \in e} \frac{1}{\omega(x)}}$
  Select at random $x \in e$ with $Prob(x) = \frac{\bar{\omega}}{\omega(x)}$
  Remove x from G and add it to C
Return C

Pitt [26] uses the primal-dual approach to prove that his algorithm is a randomized-2-approximation algorithm for Vertex Cover. Using our technique, we get:

Theorem 6: *Pitt's algorithm is a randomized-Δ_e-approximation for HVC.*

Proof: To prove correctness, we use the Randomized Local-Ratio theorem. We can regard the selection of a vertex x and its removal as if we defined $\delta(x) = \omega(x)$, and $\forall_{y \neq x}\ \delta(y) = 0$. We show that in each iteration, the distribution is Δ_e-effective. For this, we need to show that $Exp[\delta(V)] \leq r \cdot Exp[\mathrm{OPT}(\delta)]$.

Let e be the edge chosen in the first iteration, and let C^* be an optimum cover.

$$Exp[\mathrm{OPT}(\delta)] = Exp[\delta(C^*)] = \sum_{x \in e \cap C^*} p(x) \cdot \omega(x)$$

$$= \sum_{x \in e \cap C^*} \frac{\bar{\omega}}{\omega(x)} \cdot \omega(x) = \bar{\omega} \cdot |e \cap C^*| \geq \bar{\omega}$$

$$Exp[\delta(V)] = \sum_{x \in V} p(x) \cdot \omega(x) = \sum_{x \in e} \frac{\bar{\omega}}{\omega(x)} \cdot \omega(x) = \bar{\omega} \cdot |e| \leq \bar{\omega} \cdot \Delta_e$$

□

5.2 Our Modified Greedy Algorithm

The greedy algorithm for set cover [12] is a $\ln \Delta_v$-approximation. Algorithm BarEven is a Δ_e-approximation. In some families of problems (e.g., VC), Δ_e is preferable to $\ln \Delta_v$. In these cases, we would like to have a ratio of Δ_e, but we may also prefer to use the "greedy" heuristic. Can we have the best of both worlds? As we have seen, this is possible: Clarkson's modified greedy algorithms for VC can easily be modified for HVC, using the Local-Ratio theorem. But, even for unit-weight Vertex Cover, it may have a problem with robustness, due to error accumulation. The following randomized greedy modification does not "reduce weights", but rather deletes vertices:

Algorithm Random-Greedy $(G = (V, E), \omega)$

$C = \{x : \omega(x) = 0\}$ and delete C from G
While $E \neq \phi$
  Let $\bar{\omega} = \frac{1}{\sum_{x \in V} \frac{d(x)}{\omega(x)}}$
  Select at random $x \in V$ with $Prob(x) = \frac{d(x)}{\omega(x)} \cdot \bar{\omega}$
  Remove x from G and add it to C
Return C

Let C^* be an optimum cover,

$$Exp[\mathrm{OPT}(\delta)] = Exp[\delta(C^*)] = \sum_{x \in C^*} p(x) \cdot \omega(x) = \sum_{x \in C^*} d(x) \cdot \bar{\omega} \geq |E| \cdot \bar{\omega}$$

$$Exp[\delta(V)] = \sum_{x \in V} p(x) \cdot \omega(x) = \sum_{x \in V} d(x) \cdot \bar{\omega} \leq \Delta_e \cdot |E| \cdot \bar{\omega}$$

this implies:

Theorem 7: *Random-Greedy is a randomized-Δ_e-approximation for the HVC problem.*

Corollary 8: *Random-Greedy is a randomized-2-approximation for VC.*

6 Pay Only for Minimal Covers

Following Bafna, Berman and Fujito [2], we extend the Local-Ratio theorem. This will be shown to be useful in the following sections.

In the previous analyses, we used $\delta(V)$ as an upper bound for the payment in a step. This may be too much. If we knew the final cover C eventually found by the algorithm, we would know that $\delta(C)$ is the exact payment.

In practice, we can restrict the final cover C to be a minimal cover. Defined formally:
Definition A cover C is a *minimal cover* if $\forall_{x \in c}\ C \backslash \{x\}$ is not a cover. A cover C is a *minimal cover w.r.t a weight function* ω, if $\forall_{x \in C \text{ and } \omega(x)>0} C \backslash \{x\}$ is not a cover.

If we know that the final cover C is minimal w.r.t. δ, we can bound the payment ΔCOST by $\max\{\delta(C) | C \text{ is minimal w.r.t } \delta\}$.

Define $\delta : X \to R^+$ to be *minimal-r-effective* if

1. $\forall_{x \in X}\ 0 \leq \delta(X) \leq \omega(x)$
2. $\delta(C) \leq r \cdot \mathrm{OPT}(\omega)$ for all C that are minimal w.r.t. δ.

Consider the following general algorithm A:

Algorithm $A(X, f, \omega)$

Select a minimal-r-effective δ
Call algorithm $B(X, f, \omega - \delta)$ to find a cover C
Removal loop:
for each $x \in C$, if $\delta(x) > 0$ and $C \setminus \{x\}$ is a cover then $C = C \setminus \{x\}$
Return C

Theorem 9: ***The Extended Local-Ratio Theorem***
If B returns a cover C s.t. $(\omega - \delta)(C) \leq r \cdot OPT(\omega - \delta)$, then $\omega(C) \leq r \cdot OPT(\omega)$.

Proof: After the "removal loop", C is still a cover, but it is also minimal w.r.t. δ, and therefore $\delta(C) \leq r \cdot \mathrm{OPT}(\delta)$. Hence

$$\begin{aligned} \omega(C) &= (\omega - \delta)(C) + \delta(C) \\ &\leq r \cdot \mathrm{OPT}(\omega - \delta) + r \cdot \mathrm{OPT}(\delta) && \text{[by the properties of } B \text{ and } \delta] \\ &\leq r \cdot \mathrm{OPT}(\omega) && \text{[by the Decomposition Observation]} \end{aligned}$$

□

7 The Feedback Vertex Set Problem

Let $G = (V, E)$ be a simple graph with weight function $\omega : V \to R^+$. A set $F \subseteq V$ is called a Feedback Vertex Set (FVS) if $G \backslash F$ is a forest, i.e., for every cycle $C \subseteq V$, $C \cap F \neq \phi$. The FVS problem is: given a graph $G = (V, E)$ and a weight function $\omega : V \to R^+$, find a FVS, F, with minimum total weight. The FVS problem is NP-hard [16]. Furthermore, any r-approximation for FVS implies an r-approximation with the same time complexity for the VC problem (each edge in the VC instance graph can be replaced by some cycle).

This implies that a 2-approximation is the best we can hope for, as long as the ratio for VC is not improved.

The first non-trivial approximation algorithm was given by Bar-Yehuda, Geiger, Naor and Roth [7]. They present a 4-approximation for unit weights and an $O(\log n)$-approximation for the general problem. Following their paper, Bafna, Berman and Fujito [2] found the first 2-approximation. They were also the first to generalize the local ratio theorem to deal with "paying" according to a minimal cover. Their approach has helped to deepen our understanding of approximations for covering problems.

Following their approach, Becker and Geiger [9] present a relatively simple 2-approximation. Following all these, Chudak, Goemans, Hochbaum and Williamson [11] explained all of these algorithms in terms of the primal-dual method. They also gave a new 2-approximation algorithm, which is a simplification of the Bafna et al. algorithm. Using the Extended Local Ratio theorem, we can further simplify the presentation and proof of all these three algorithms, [2, 9, 11]. Let us illustrate this for the Becker-Geiger algorithm.

The main idea behind Becker and Geiger's algorithm is the weighted graph we call a degree-weighted graph. A *degree-weighted graph* is a pair (G, ω) s.t. each vertex v has a weight $\omega(v) = d(v) > 1$.

Theorem 10*: Every minimal FVS in a degree-weighted graph is a 2-approximation.*

Proof: Let C be a minimal cover, and C^* an optimum cover. We need to prove $\omega(C) \leq 2 \cdot \omega(C^*)$. It is enough to show that

$$\sum_{x \in C} \omega(x) \leq 2 \cdot \sum_{x \in C^*} \omega(x)$$

by definition, it is enough to show that

$$\sum_{x \in C} d(x) \leq 2 \cdot \sum_{x \in C^*} d(x).$$

This is proved in [9]. A simple proof also appears in [11]. □

Corollary 11*: If $G = (V, E)$ is a graph with $\forall_{v \in V}\ d(v) > 1$, and $\delta : V \to R^+$ satisfies $\forall_{v \in V}\ 0 \leq \delta(v) = \epsilon \cdot d(v) \leq \omega(v)$ for some ϵ, then δ is 2-effective.*

Proof: We only need to show that for every minimal cover C, $\delta(C) \leq 2 \cdot \mathrm{OPT}(\delta)$. This is immediate from Theorem 10. □

In order to guarantee termination, we choose $\epsilon = \min_{v \in V} \frac{\omega(v)}{d(v)}$.

Algorithm BeckerGeiger $(G = (V, E), \omega)$

If G is a tree return ϕ
If $\exists_{x \in V}\ d(x) \leq 1$ return BeckerGeiger$(G \backslash \{x\}, \omega)$
If $\exists_{x \in V}\ w(x) = 0$ return $\{x\}$+BeckerGeiger$(G \backslash \{x\}, \omega)$
Let $\epsilon = \min_{v \in V} \frac{\omega(v)}{d(v)}$
Define $\delta : \forall_{x \in V}\ \delta(x) = \epsilon \cdot d(x)$
$C =$ BeckerGeiger$(G, \omega - \delta)$
Removal loop:
for each $x \in C$, if $\delta(x) > 0$ and $C \backslash \{x\}$ is a cover then $C = C \backslash \{x\}$
Return C

Theorem 12: *Algorithm BeckerGeiger is a 2-approximation for FVC.*

Proof: The algorithm terminates, since in each recursive call, at least one more vertex weight becomes zero. Therefore we can use induction. The base of induction is trivial. Let us assume inductively that the recursive call returns C s.t.

$$(\omega - \delta)(C) \leq 2 \cdot \mathrm{OPT}(\omega - \delta).$$

Since, by Corollary 11, δ is 2-effective, we can use the Extended Local-Ratio theorem to conclude that $\omega(C) \leq 2 \cdot \mathrm{OPT}(\omega)$. □

8 Generalized Steiner Forest and Related Problems

Goemans and Williamson [17] have recently presented a general approximation technique for covering cuts of vertices with edges. Their framework includes the Shortest Path, Minimum-cost Spanning Tree, Minimum-weight Perfect Matching, Traveling Salesman, and Generalized Steiner Forest problems. All these problems are called *Constraint Forest Problems.*

To simplify the presentation, let us choose the Generalized Steiner Forest problem. We are given a graph $G = (V, E)$, a weight function $\omega : E \rightarrow R^+$, and a collection of terminal sets $T = \{T_1, T_2, \ldots, T_t\}$, each a subset of V. A subset of edges $C \subseteq E$ is called a Steiner-cover if, for each T_i, and for every pair of terminals $x, y \in T_i$, there exists a path in C connecting x to y. The Generalized Steiner Forest problem is: given a triple (G, T, ω), find a Steiner-cover C with minimum total weight $\omega(C)$.

In order to get a 2-effective weight function, let us define an edge-degree-weighted graph.

An *edge-degree-weighted graph* is a triple (G, T, ω) s.t.

$$\forall_{e \in E}\ \omega(e) = d_T(e) = |\{x \in e : \exists_i\ x \in T_i\}| \quad .$$

Theorem 13: *Every minimal cover in an edge-degree-weighted graph is a 2-approximation.*

Proof: The proof is an elementary exercise, see, e.g., [18]. □

So now, given (G, T, ω), we can easily compute a 2-effective weight function $\delta(e) = d_T(e) \cdot \epsilon$ for all $e \in E$, and in order to guarantee termination, we select $\epsilon = \min\{\omega(e)/d_T(e) : d_T(e) \neq 0\}$.

Now we can proceed as follows:

"Shrink" all pairs $e = \{x, y\}$ s.t. $(\omega - \delta)(e) = 0$.

Recursively, find a minimal 2-approximation Steiner-cover, C. Add all "shrunken" edges to the cover C. In order to guarantee the minimality property, apply the following removal loop:

For each "shrunken" edge e do: if $C \setminus \{x\}$ is a Steiner-cover, delete e from C

Remarks: Two interesting special cases mentioned by Goemans and Williamson in [17]:

- The Minimum Spanning Tree problem: If we reduce ϵ from all the edges in the graph, we reduce the optimum by exactly $(n-1)\epsilon$. By taking only minimal solutions we will pay only $(n-1)\epsilon$ for this reduction. Thus we get an optimization algorithm, which is, in fact, Kurskal's algorithm [23] (Note that we also get this algorithm by using the algorithm for the Generalized Steiner Forest with $t=1$ and $T_1 = V$).
- The (s,t) Shortest Path problem: If we reduce ϵ from all edges connected to s, we reduce the optimum by exactly ϵ. By taking only minimal solutions we will pay only ϵ for this reduction. Thus we get Dijkstra's algorithm [14]. By using the algorithm for the Generalized Steiner Forest with $t=1$ and $T_1 = \{s,t\}$ we get the bidirectional version of Dijkstra algorithm.

9 A Generic r-Approximation Algorithm

We now present a generic r-approximation algorithm that can be used to generate all the deterministic algorithms presented in this paper.

The weight function δ is called ω-tight if $\exists_{x \in X}\ \delta(x) = \omega(x) > 0$.

Algorithm Cover (X, f, ω)

If the set $\{x \in X : \omega(x) = 0\}$ is a cover then return this set.
Select ω-tight minimal-r-effective $\delta : X \to R^+$
Recursively call Cover $(X, f, \omega - \delta)$ to get a cover C
Removal loop:
for each $x \in C$, if $\delta(x) > 0$ and $C \backslash \{x\}$ is a cover then $C = C \backslash \{x\}$
Return C

Theorem 14: *Algorithm Cover is an r-approximation.*

Proof: Define $X^+ = \{x \in X : \omega(x) > 0\}$. Since δ is ω-tight, the size of X^+ decreases at each recursive call. This implies that the total number of recursive calls is bounded by $|X|$. We can now prove the theorem using induction on $|X^+|$.

Basis: $|X^+| = 0$, hence $\omega(X) = 0$.

Step: We can consider the recursive call as the algorithm B in the Extended Local-Ratio theorem, Theorem 9. By the induction hypothesis, this recursive call, Cover $(X, f, \omega - \delta)$, satisfies B's requirement in the theorem, i.e., $(\omega - \delta)(X) \leq r \cdot \mathrm{OPT}(\omega - \delta)$. Since δ is minimal-r-effective, it remains to show that C is a minimal cover w.r.t δ. This follows from the last step of algorithm Cover. □

Acknowledgment
We would like to thank Yishay Rabinovitch, Joseph Naor and Dror Rawitz for helpful discussions, Avigail Orni and Hadas Heier for their careful reading and suggestions, and Yvonne Sagi for typing.

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized steiner problem in networks. *Proc. 23rd ACM Symp. on Theory of Computing*, 134–144, 1991.

[2] V. Bafna, P. Berman, and T. Fujito. Constant ratio approximation of the weighted feedback vertex set problem for undirected graphs. *ISAAC '95 Algorithms and Computation*, (1004):142–151, 1995.

[3] R. Bar-Yehuda and S. Even. A linear time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2:198–203, 1981.

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

[5] R. Bar-Yehuda. A linear time 2-approximation algorithm for the min clique-complement problem. *Technical Report CS0933, Technion Haifa*, May 1998.

[6] R. Bar-Yehuda. Partial vertex cover problem and its generalizations. *Technical Report CS0934, Technion Haifa*, May 1998.

[7] R. Bar-Yehuda, D. Geiger, J. Naor, and R. Roth. Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and bayesian inference. *Accepted to SIAM J. on Computing*, 1997.

[8] R. Bar-Yehuda and D. Rawitz. Generalized algorithms for bounded integer programs with two variables per constraint. *Technical Report CS0935, Technion Haifa*, May 1998.

[9] A. Becker and D. Geiger. Approximation algorithms for the loop cutset problem. *Proc. 10th Conf. on Uncertainty in Artificial Intelligence*, 60–68, 1994.

[10] D. Bertsimas and C.P. Teo. From valid inequalities to heuristics: a unified view of primal-dual approximation algorithms in covering problems. *Proceedings of the 6th Annual Symposium on Discrete Algorithms*, 102–111, 1995.

[11] F. Chudak, M. Goemans, D. Hochbaum, and D. Williamson. A primal-dual interpretation of recent 2-approximation algorithms for the feedback vertex set problem in undirected graphs. *Unpublished*, 1996.

[12] V. Chvatal. A greedy heuristic for the set-covering problem. *Math. of Oper. Res.*, 4(3):233–235, 1979.

[13] K. Clarkson. A modification of the greedy algorithm for the vertex cover. *Info. Proc. Lett.*, 16:23–25, 1983.

[14] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269-271, 1959.

[15] T. Fujito. A unified local ratio approximation of node-deletion problems. *ESA, Barcelona, Spain*, 167–178, September 1996.

[16] M. Garey and D. Johnson. *Computers and Intractability*. W.H. Freeman, 1979.

[17] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.

[18] M. Goemans and D. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. *Approximation Algorithms for NP-Hard Problems*, 4, 1996.

[19] D. S. Hochbaum. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. *Chapter 3 in Approximation Algorithms for NP-Hard Problems, PWS Publication Company*, 1997.

[20] D. S. Hochboum. Approximating clique and biclique problems. *Manuscript, UC berkeley*, 1997.

[21] D. S. Hochboum. The t-vertex cover problem: Extending the half integrality framework with budget constraints. *APPROX'98 1st International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 111-122, 1998.

[22] D. S. Hochbaum, N. Megido, J. Naor and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, 62:69–83, 1993.

[23] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of th American Mathematical Society*, 7:48-50, 1956.

[24] G. Nemhauser and J. L.E. Trotter. Vertex packings: structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[25] V. T. Paschos. A survey of approximately optimal solutions to some covering and packing problems. *ACM Computing Surveys*, 29(2):171–209, June 1997.

[26] L. Pitt. Simple probabilistic approximation algorithm for the vertex cover problem. *Technical Report, Yale*, June 1984.

[27] V. Vazirani. *Approximation algorithms, College of Computing, Georgia Institute of Technology*, 1997.

[28] D. Williamson, M. Goemans, M. Mihail, and V. Vazirani. A primal-dual approximation algorithm for generalized steiner network problems. *Combinatorica*, 15:435–454, 1995.