A Linear Time 2-Approximation Algorithm for the Min Clique-Complement Problem

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

May 11, 1998

Abstract

Using the local-latio technique, we present a linear time 2-approximation algorithm for the Min Clique-Complement problem.

Keywords: Approximation Algorithm, Local Ratio, Covering Problems, Vertex Cover, Min Clique-Complement problem.

1 Introduction

We are given a simple graph $G = (V, E)$, for each edge $e \in E$ a weight $\omega(e) \geq 0$. Define:
$\omega(E') = \sum_{v \in V'} \omega(e)$ for $E' \subseteq E$.

A set $C \subseteq E$ is called a *clique-complement cover* (or just a *cover*) if the set of edges $E \setminus C$ induces a complete graph. The Min Clique-Complement problem is: given $G = (V, E)$, and for every edge $e \in E$ a weight $\omega(e) \geq 0$, find a cover with a minimum total weight. A cover C^* is optimal if $\omega(C^*) \leq \omega(C)$ for all covers C. A cover C is called an *r-approximation*, if $\omega(C) \leq r \cdot \omega(C^*)$.

An algorithm A is called an r-approximation algorithm if for all instances G, ω, A returns an r-approximate cover.

While the Maximum Clique problem is known to be hard to approximate, as shown by Hastad [3], Hochboum [4] has developed a 4-approximation for the Min Clique-Complement problem. In this paper we further improve Hochboum's result. We observe that 2-approximation is easy to get be reduction to the Vertex Cover Problem. Our conribution is mainly in the time complexity. Using the local-ratio technique see e.g. [2], [1], we obtain a linear time

2-approximation. In the second section, we give the general 2-approximation algorithm. In the third section, we show a linear time implementation.

2 A 2-Approximation Algorithm

The main idea: let e_1 and e_2 be two edges having at least two non-neighbor endpoints, hence e_1 and e_2 cannot participate in the same complete subgraph. This implies that either e_1 or e_2 has to be eliminated, i.e., for every cover $C \subseteq E$, $C \cap \{e_1, e_2\} \neq \phi$.

At this point, one may suggest, to make a reduction to the Vertex Cover Problem to get the 2-approximation desired. However, such a reduction is time and space expansive.

Consider the following subroutine:

Subroutine Edge-Local-Ratio(e_1, e_2, ω)
  Let $\epsilon = min\{\omega(e_1), \omega(e_2)\}$
  $\omega(e_1) = \omega(e_1) - \epsilon$
  $\omega(e_2) = \omega(e_2) - \epsilon$

Two edges e_1, e_2 are called *conflicting* if $\omega(e_1) > 0$ and $\omega(e_2) > 0$ and they have non-neighbor endpoints. Consider the following general algorithm CCC1 (Clique-Complement-Cover-1):

Algorithm CCC1$(G = (V, E), \omega)$
  While $\exists$ conflicting edges e_1, e_2
    Edge-Local-Ratio(e_1, e_2, ω)
  Clique $= \{v : \ \exists_{e \in E} \ v \in e$ and $\omega(e) > 0\}$
  Return $C = \{\{x, y\} \in E : \ x \notin$ *Clique* or $y \notin$ *Clique*$\}$

Theorem 2.1 *Algorithm CCC1 is a 2-approximation.*

Proof. We need to prove

(i) C is a cover.

(ii) C is a 2-approximation.

(i) To prove that C is a cover, it is sufficient to show that *Clique* induces a complete graph. Let $x, y \in$ *Clique*. Let e_x, e_y be incident edges to x and y respectively, s.t. $\omega(x) > 0$ and $\omega(y) > 0$. By the termination condition of the algorithm, we know that e_x and e_y are not conflicting. This implies that x and y are neighbors. Hence every pair of vertices in *Clique* are connected.

(ii) The proof proceeds by induction on the number of iterations (This number is obviously finite).

Base: 0 iterations means $\omega(C) = 0$.

Step: Define $\omega_1(e_1) = \omega_1(e_2) = \epsilon$, and $\forall_{e \notin \{e_1, e_2\}} \omega(e) = 0$. Also define $\omega_2(\cdot) = \omega(\cdot) - \omega_1(\cdot)$. Let C^*, C_1^*, and C_2^* be optimal covers with respect to ω, ω_1, and ω_2.

$$\begin{aligned}
\omega(C) &= \omega_1(C) + \omega_2(C) && \text{[by definition]} \\
&\leq 2 \cdot \omega_1(C_1^*) + 2\omega_2(C_2^*) && [e_1, e_2 \text{ conflicting pair and induction hyp.}] \\
&\leq 2 \cdot \omega_1(C^*) + 2\omega_2(C^*) && [C_1^*, \text{ and } C_2^* \text{ are covers of } \omega_1 \text{ and } \omega_2] \\
&\leq 2 \cdot \omega(C^*) && [\omega = \omega_1 + \omega_2]
\end{aligned}$$

□

One can easily see how to implement this in $O(|E|^2)$ time or even in $O(|E| + |V|^2)$ time. The next section gives a linear time implementation.

3 A Linear Time 2-Approximation

In order to implement Algorithm CCC1 in linear time, consider the following subroutine, which receives non-adjacent vertices x, y, and applies the Edge-Local-Ratio subroutine until one of the vertices is attached only to zero-weight edges. The following subroutine assumes an incidence-list data structure for G.

```
Subroutine Vertex-Local-Ratio(x, y, ω, G)
   E_x = the set of non-zero-weight incident edges of x
   E_y = the set of non-zero-weight incident edges of y
   e_x = POP(E_x) ; e_y = POP(E_y) ; Edge-Local-Ratio(e_x, e_y, ω)
   While E_x ≠ φ and E_y ≠ φ
      If ω(e_x) = 0
         e_x = POP(E_x)
      Else
         e_y = POP(E_y)
      Edge-Local-Ratio(e_x, e_y, ω)
```

We use this subroutine in the following algorithm:

Algorithm CCC2$(G=(V,E),\omega)$

$U=V$
While $U \neq \phi$
  $x = POP(U)$
  Move all neighbors of x from U to U'
  While $U \neq \phi$ and $\omega(E(x)) > 0$
    $y = POP(U)$
    Vertex-Local-Ratio(x,y,ω,G)
    If $\omega(E(y)) > 0$ PUSH y into U
  Move U' back to U

Theorem 3.2 *Algorithm CCC2 is a linear time 2-approximation algorithm.*

Proof. It is obvious that CCC2 is an implementation of CCC1, since it only reduces weights according to Subroutine Edge-Local-Ratio(e_1,e_2,ω), where e_1,e_2 are conflicting edges. Also, when the algorithm terminates, there are no two conflicting edges. This implies that the algorithm is a 2-approximation, by Theorem 2.1.

To see that the time complexity is $O(|E|)$, observe that all edge weight subtractions done by Edge-Local-Ratio can be charged to the edge e that is "deleted", i.e., $\omega(e)=0$. This implies that all calls to Vertex-Local-Ratio are also "for free". Moving neighbors of x from U to U' and back costs a total of $\sum_{v\in V} d(v) = O(|E|)$, and the rest is trivial. □

Acknowledgment
We would like to thank Avigail Orni and Dror Rawitz for their careful reading and suggestions.

References

[1] R. Bar-Yehuda. One for the price of two: A unified approach for approximating covering problems, to appear. *APPROX'98 1st International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, 1998.

[2] 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.

[3] J. Hastad. Clique is hard to approximate within $n^{1+\epsilon}$. *Proc 37th IEEE Symp. on Foundation of Computer Science*, pages 627–636, 1996.

[4] D. Hochbaum. Approximating clique and biclique problems. *Unpublished*, 1997.