Min Sum Clustering with Penalties

*Refael Hassin† Einat Or ‡

Abstract

Traditionally, clustering problems are investigated under the assumption that all objects must be clustered. A shortcoming of this formulation is that a few distant objects, called *outliers*, may exert a disproportionately strong influence over the solution. In this work we investigate the k-min-sum clustering problem while addressing outliers in a meaningful way.

Given a complete graph $G = (V, E)$, a weight function $w : E \to I\!N_0$ on its edges, and $p \to I\!N_0$ a penalty function on its vertices, the PENALIZED k-MIN-SUM PROBLEM is the problem of finding a partition of V to $k+1$ sets, $S_1, \ldots, S_{k+1}$, minimizing $\sum_{i=1}^{k} w(S_i) + p(S_{k+1})$, where for $S \subseteq V$ $w(S) = \sum_{e=\{i,j\}\subseteq S} w_e$, and $p(S) = \sum_{i\in S} p_i$.

Our main result is a randomized approximation scheme for the metric version of the penalized 1-min-sum problem, when the ratio between the minimal and maximal penalty is bounded. For the metric penalized k-min-sum problem where k is a constant, we offer a 2-approximation.

Keywords: Min sum clustering; outliers; randomized approximation scheme

1 Introduction

1.1 Clustering with penalties

Clustering problems such as k-MEDIAN, k-CENTER and k-MIN-SUM are widely studied in operations research and computer science. Traditionally these problems are investigated under the assumption that all objects must be clustered. A significant shortcoming of this formulation is that a few very distant objects, called *outliers*, may exert a disproportionately strong influence over the solution. In this work we investigate the k-MIN-SUM clustering problem while addressing outliers in a meaningful way.

Given a complete graph $G = (V, E)$ with $|V| = n$, and a weight function $w : E \to I\!N_0$, on its edges, let $w(S) = \sum_{e=\{i,j\}\subseteq S} w_e$. The k-MIN-SUM problem is to compute a partition of V to k sets, $\{S_1, \ldots, S_k\}$, minimizing $\sum_{i=1}^{k} w(S_i)$. It is NP-hard even for $k = 2$, and cannot be approximated up to any constant for $k \geq 3$ [21]. An $O(\log n)$-approximation

*An early version of this paper appeared in *Proceedings of ESA 2005*, LNCS 3669, 167-178.

†Department of Statistics and Operations Research, Tel-Aviv University, Tel-Aviv 69978, Israel. Tel: +97236409281 Email: hassin@math.tau.ac.il

‡Department of Statistics and Operations Research, Tel-Aviv University, Tel-Aviv 69978, Israel.

algorithm for $k = 2$ was given by Garg et al [14] . However if w is a metric, then there is a randomized PTAS for any fixed k by Fernandez de la Vega et al. [8]. In the following we wish to generalize the k-MIN-SUM problem and allow outliers that are not clustered. Let $p \rightarrow I\!N_0$ denote the penalty function on the vertices of G. For $S \subseteq V$ let $p(S) = \sum_{i \in S} p_i$. The PENALIZED k-MIN-SUM PROBLEM is the problem of finding a partition of V to $k+1$ sets, $\{S_1, \ldots, S_{k+1}\}$ minimizing $\sum_{i=1}^{k} w(S_i) + p(S_{k+1})$. We denote this problem by kPMS, and use PMS to denote 1PMS. The formulation of kPMS makes no assumptions regarding the number of outliers, the outliers are determined by the clustering procedure.

PMS is a partition of the vertices to two sets, one constitutes the cluster while the other contains the non clustered points. MAX-CUT, MIN-BISECTION and 2-MIN-SUM are three examples of NP-hard problems that require a partition of a set into two sets, optimizing a function of the weight on the edges. PMS is different since the target to be optimized includes the weight on the **edges** in the cluster, and the weight on the **vertices** left out of the cluster.

1.2 Related work

PMS is related to the MINIMUM l-DISPERSION PROBLEM, which is the problem of finding a subset $V' \subset V$ of predetermined cardinality l that minimizes $w(V')$. The maximization version of this problem has an $O(n^{\delta})$ approximation for some $\delta < \frac{1}{3}$ by Feige, Kortsarz and Peleg [9], and the metric case has a $\frac{1}{2}$-approximation [17].

A modeling of outliers for the location problems k-MEDIAN, and UNCAPACITATED FACILITY LOCATION is presented by Charikar et al [3] . The outliers are integrated into the objective function by penalizing the clients who are not served. A 4-approximation is presented for the metric k-MEDIAN with penalties, and a 3-approximation is obtained for the UNCAPACITATED FACILITY LOCATION problem. For the UNCAPACITATED FACILITY LOCATION an LP rounding technique with the same bound is presented by G. Xu and J. Xu in [22]. A constant factor approximation for CORRELATION CLUSTERING WITH PENALTIES is given by Aboud and Rabani [1].

Chen [4] presents a constant factor approximation for the k-MEDIAN WITH OUTLIERS PROBLEM, where it is required to remove at most m points (the outliers) from a data set, such that the cost of the optimal k-median clustering of the remaining data is minimal. Scheduling problems with outliers are considered by Gupta et al [11] and other papers referenced there.

Our work borrows techniques from several papers: Randomized PTAS for metric MAX-CUT by Fernandez de la Vega and Kenyon [6], metric MIN-BISECTION by Fernandez de la Vega, Karpinski and Kenyon [7], metric 2-MIN-SUM by Indyk [18, 8], and metric k-MIN-SUM for fixed k by Fernandez de la Vega et al [8], were presented in recent years. In [8] the approximation algorithm for 2-MIN-SUM uses the PTAS given in [6] for MAX-CUT when the clusters are not well separated, i.e the weight of the cut is not much greater than the weight of the clusters. When the clusters are well separated, it is sufficient to find a representative from each cluster and use it to estimate the distance between a vertex and each one of the clusters. The estimate is the distance between the vertex and the representative of the

cluster, multiplied by the size of the cluster. Using the separability of the clusters, this estimation is sufficient to produce a PTAS.

In [7] the MIN-BISECTION is addressed. In this case the two sides of the bisection are not well separated, since the cut is minimized, and the problem of good estimation of the distance of a vertex to each side of the bisection arises. A natural approach to the problem is to use a sample from each side of the bisection as the basis of the estimation. It is proved that a sampling method referred to as *metric sampling* gives a good estimation.

1.3 Our contribution

We prove that PMS is NP-hard even if w is a metric and all penalties are equal. With general w and equal penalties, PMS is at least as hard to approximate as VERTEX COVER.

A 2-approximation for PMS follows from the approximation framework presented by Hochbaum [15]. We observe that a faster 2-approximation algorithm can be obtained by using the same LP relaxation, but rounding it via a primal-dual algorithm.

Our main result is a randomized approximation scheme for the metric PMS where the ratio between the minimal and the maximal penalty is bounded, for example if all penalties are equal or there is a constant number of different penalties. The algorithm is based on methods used to approximate MIN-BISECTION and 2-MIN-SUM [7, 8]. While the approach in [8] is a PTAS for metric PMS when the cluster includes most of the vertices, it gives poor results if the cluster is smaller. The approach in [7] is the basis for a PTAS for metric PMS where the cluster and the set of non-clustered points are both large, but it gives poor approximation if one of the parts is small. Therefore we present a combination of the two approaches. For the metric kPMS where k is a constant, we offer a 2-approximation by generalizing an algorithm of Guttmann-Beck and Hassin [12] for min-sum p-clustering.

The paper is organized as follows. In Section 2 we present hardness results and lower bounds. In Section 3 we give approximation algorithms for PMS. In Section 4 we give a PTAS for metric PMS with uniform penalties. In Section 4.1 we describe a 2-approximation for metric kPMS with fixed k.

2 Hardness results and lower bounds

Theorem 2.1 *PMS is NP-hard even if w is a metric and $p_v = p$ for every $v \in V$.*

Proof: Reduction from l-CLIQUE: Given a graph $G = (V, E)$ and a number l, is there a clique of size l in the graph? We construct the following instance of PMS. We complete G and set $w_e = 1$ for every $e \in E$ and $w_e = 2$ for every $e \notin E$. Let $p_i = l - 0.5$ for every $i \in V$. Let opt denote the optimal solution value of PMS.

The optimal cluster has at most l vertices, because in a cluster greater than l, each vertex increases the value of the solution by at least l by paying for its edges in the cluster, instead of paying $l - \frac{1}{2}$ by taking it out of the cluster.

We will show that if there is a clique of size l then opt $\leq \frac{1}{2}l(l-1)+(n-l)(l-\frac{1}{2})$ whereas if such a clique does not exist in G, opt $> \frac{1}{2}l(l-1)+(n-l)(l-\frac{1}{2})$.

Denote by c^* the value of a solution whose cluster is an l-clique of G, if one exists. Then, $c^* = \frac{1}{2}l(l-1)+(n-l)(l-\frac{1}{2})$, and the first part of the claim is proved.

Let C be a solution with $|C| = m \leq l$ and value c. Then,

$$c - c^* \geq \left[\binom{m}{2} + (n-m)\left(l-\frac{1}{2}\right)\right] - \left[\binom{l}{2} + (n-l)\left(l-\frac{1}{2}\right)\right] = \frac{1}{2}(m-l)^2 \geq 0.$$

If there is no l-clique in G then either $m < l$ and the second inequality is strict, or $m = l$ and the cluster contains at least one edge of weight 2, so that the first inequality is strict. In both cases, $c > c^*$. ∎

Theorem 2.2 *PMS is at least as hard to approximate as* VERTEX COVER *(VC).*

Proof: Given a graph $G = (V, E)$, instance of VERTEX COVER, we construct an instance of PMS such that after excluding clusters with very high cost, the feasible sets for VC and PMS coincide. We complete G and set $w_e = |V|$ for every $e \in E$ and $w_e = 0$ for every $e \notin E$, and set $p_i = 1$ for every $i \in V$. Clearly, PMS chooses an independent set C of G to avoid paying $|V|$ for an edge, and its cost is therefore the size of the vertex cover $V \setminus C$. Therefore, after excluding the clusters which contain expensive edges, the feasible solutions for the PMS instance are exactly the complements of a vertex cover, and their cost is the size of the vertex cover. ∎

We note that Dinur and Safra [5] proved that it is NP-hard to approximate minimum vertex cover within any factor smaller than $10\sqrt{5}-21 \approx 1.3606$. Khot and Regev [19] proved that vertex cover is hard to approximate within any constant factor smaller than 2, assuming the unique games conjecture.

3 2-approximation for PMS

Let $x_e = 1$ ($x_e = 0$) if the edge e is (is not) in the cluster, and $y_i = 1$ ($y_i = 0$) if the vertex i is not in (is in) the cluster. PMS is

$$\begin{array}{lll} \text{Minimize} & \sum_{e\in E} w_e x_e + \sum_{i\in V} p_i y_i & \\ & y_i + y_j + x_e \geq 1 & \forall e = \{i,j\} \in E, \\ & x_e \in \{0,1\} & \forall e = \{i,j\} \in E, \\ & y_i \in \{0,1\} & \forall i \in V. \end{array}$$

We note that this problem, after changing the objective from minimizing the *cost*, to maximizing the *profit*, is the PRIZE COLLECTING VERTEX COVER PROBLEM introduced in

[16]. The linear programming relaxation, denoted LP-primal, is:

$$\begin{array}{lll} \text{Minimize} & \sum_{e\in E} w_e x_e + \sum_{i\in V} p_i y_i & \\ & y_i + y_j + x_e \geq 1 & \forall e = \{i,j\} \in E, \\ & x_e \geq 0 & \forall e = \{i,j\} \in E, \\ & y_i \geq 0 & \forall i \in V. \end{array}$$

LP-primal has only half integral basic solutions [15], and a 2-approximation algorithm is presented in [15] for a family of problems with the same type of constraints with time complexity $O(mn\log(\frac{n^2}{m}))$ where $m = |E|$. We present a 2-approximation algorithm, denoted by PD, with time complexity $O(m)$.

Let $\delta(i) = \{e \in E | e = \{i,j\} \in E,\ j \in V\}$. The dual of the relaxation, LP-dual, is:

$$\begin{array}{lll} \text{Maximize} & \sum_{e\in E} z_e & \\ & z_e \leq w_e & \forall e \in E, \\ & \sum_{e\in\delta(i)} z_e \leq p_i & \forall i \in V, \\ & z_e \geq 0 & \forall e \in E. \end{array}$$

A *maximal solution* to LP-dual is a feasible solution for which any increase in a variable z_e, $e \in E$, results in a non feasible solution. Algorithm PD given in Figure 3 has time complexity $O(m)$ since a maximal solution to LP-dual can be found by setting initially bounds $u_i = p_i$, and then scanning V in an arbitrary order, setting for $e = \{i,j\}$ $z_e := \min\{u_i, u_j, w_e\}$ and updating $u_i := u_i - z_e$ and $u_j := u_j - z_e$.

PD
input
1. A complete graph $G = (V, E)$.
2. A function $w : E \to I\!N$.
3. A function $p : V \to I\!N$.
output
A cluster C.
begin
Find a maximal solution, $\hat{R}$, to the dual problem.
$C := V \setminus \{i \in V | \sum_{e\in\delta(i)} \hat{R}_e = p_i\}$.
return C
end PD

Figure 1: Algorithm PD

Claim 3.1 *Algorithm PD returns a 2-approximation to PMS.*

Proof: Denote the value of the approximation by $apx = \sum_{e=\{i,j\}\in C} w_e + \sum_{i\in V\setminus C} p_i$, the value of the dual relaxation by $dual = \sum_{e\in E} z_e$, and the optimal solution value by opt. Note that apx is the sum of p_i values over $i \in V$ satisfying equality in the constraint $\sum_{e\in\delta(i)} z_e \leq p_i$,

and weights w_e of edges e whose both ends satisfy strict inequality in the above constraint. Consider an edge $e = \{i, j\} \in E$. If i and j are in $V \setminus C$, then $\sum_{e \in \delta(i)} z_e = p_i$, and $\sum_{e \in \delta(j)} z_e = p_j$, and hence z_e is charged in apx twice, once in p_i and once in p_j. If $i \in V \setminus C$ and $j \in C$ then z_e is charged in apx only once in p_i. If $i \in C$ and $j \in C$ then by the maximality of the solution $z_e = w_e$, and thus R_e is charged in apx only once. We get: apx $\leq 2 \sum_{e \in E} z_e = 2dual \leq 2$opt. ∎

4 PTAS for metric w and uniform penalties

In this section we assume that w is a metric and $p_i = p$ for every $i \in V$.

We consider an optimal solution (C^*, P^*) and use the following notation:
opt - the value of the optimal solution.
apx - the value of the approximation.
C^* - the cluster in the optimal solution, and $P^* = V \setminus C^*$.
For $A, B \subset V$ and $u \in V$:
$w(A, B) = \sum_{a \in A} \sum_{b \in B} w(a, b)$, $w(u, B) = w(\{u\}, B)$,
$w(A) = \frac{1}{2} w(A, A)$,
$d_u = w(u, V)$, and $D_A = w(A, V)$.
Note that $D_A = w(A, V \setminus A) + w(A, A) = \sum_{u \in A} w(u, V)$.

Definition 4.1 *[7] A* metric sample of U *of size t is a random sample $\{u_1, \ldots, u_t\}$ of U with replacement, where each u_i is obtained by picking a point $u \in U$ with probability $\frac{d_u}{D_U}$.*

Note that for any metric, the probability of selecting any given vertex is at least $\frac{1}{2(n-1)}$. Consequently, it can be shown that with high probability, the number of distinct vertices in a metric sample of size $t = o(n)$ is at least $\frac{t}{2}$. Following [7] we simplify the presentation and consider the metric sample as a set and not a multiset.

For $|C^*| \leq \frac{1}{\epsilon^2}$ or $|P^*| \leq \frac{1}{\epsilon^2}$, the problem can be solved by exhaustive search in polynomial time, hence in the following we assume

$$|P^*|, |C^*| > \frac{1}{\epsilon^2}. \tag{1}$$

An intuitive approach to the problem would be to take a random sample of C^* , use it to estimate the distance of each vertex from C^*, and then form a cluster from the vertices closest to the sample. This approach fails, in general, because the distance between the points inserted to the cluster is not estimated and it is part of the weight of the cluster.

In several steps of our proposed algorithm, we assume that a certain unknown value can be used in the computations, assuming that an exhaustive search is applied to a polynomial number of possible values. For example, to compute a value j such that $(1+\epsilon)^j \leq D_{C^*} < (1+\epsilon)^{j+1}$, we use the fact that $D_{C^*} \leq w(V)$ and search over $O(\log_{1+\epsilon} w(V))$ values.

Algorithm Al is presented in Figure 2. It calls the two algorithms UC and BC presented in Figures 3 and 4, respectively. The input to each of these algorithms consists of the complete graph $G = (V, E)$ with $|V| = n$, a metric $w = w(i, j)$ defined on $(i, j) \in E$, the penalty p for not including a vertex in the cluster, and a positive constant ϵ. Algorithm Al produces three solutions which guarantee good approximation in each of the following three cases.

If $|C^*| \leq \epsilon n$ then $C = \emptyset$ is a $(1 + 2\epsilon) - approximation$, as proved in Theorem 4.2 below.

For the case $|P^*| \leq \epsilon n$ we can use a method presented in [18, 8] for the METRIC 2-MIN SUM PROBLEM. In general, the method in [8] is to find a representing vertex of the cluster, use it to estimate the distance between every vertex and the cluster, and add the close vertices to the cluster. This approach works because the number of vertices misplaced in the cluster is bounded by $|P^*| \leq \epsilon n$. This method is used in [8] for METRIC 2-MIN SUM PROBLEM for the case where one of the clusters is much greater than the other and the clusters are well separated, that is, the maximum cut is much greater than the weight of the clusters. In PMS, C^* and P^* are not necessarily well separated, but as will be proved in Theorem 4.4 below, an algorithm based on this method is a $PTAS$ when $|P^*| \leq \epsilon n$. We denote this part of our algorithm by UC.

If $|C^*| > \epsilon n$ and $|P^*| > \epsilon n$ we encounter the following difficulties: Even a large random sample of C^* does not estimate, with sufficient accuracy, the distance $w(v, C^*)$. This problem was addressed and solved in [7] by taking a metric sample which enables to estimate $w(v, C^*)$ accurately. But good estimation of $w(v, C^*)$ is not enough. Consider the following instance of PMS: $V = A \cup B$ where $|A| = |B| = n$, all distances in A and between A and B are 1, the distances between points of B are 2, and $p = n - 1$. The optimal solutions are $C^* = A$, and $C^* = (A \setminus \{v\}) \cup \{u\}$ for every $v \in A$ and $u \in B$. Note that for every sample T of A, $w(v, T) = w(u, T)$ where $v \in A$ and $u \in B$. Adding to the cluster points closer to the sample may lead to adding only points of B, resulting in a poor approximation. The distance **between** the points added to the sample should also be considered. We consider the distances between the added points using the *hybrid partition method* presented in [10] and used in [7]. The use in [7] is for the creation of a cut, whereas we create a cluster and hence the analysis is different.

Our algorithm for the case where $C^* > \epsilon n$ and $P^* > \epsilon n$ is denoted as BC. It BC begins by taking a metric sample T of size $O(\frac{\ln n}{\epsilon^4})$ from V, by exhaustive search finding $T^* = C^* \cap T$, and using it to estimate $w(v, C^*)$ for every $v \in V$. Let $\hat{e}_v$ denote the estimate of $w(v, C^*)$, and let C denote the cluster returned by Algorithm BC. We consider only the vertices with $\hat{e}_v \leq (1 + \epsilon)p$ as candidates for C. We partition these vertices into the following two sets, $C_{-\epsilon} = \{v \in V |\ \hat{e}_v \leq p(1 - \epsilon)\}$ and $C_{\pm\epsilon} = \{v \in V |\ p(1 - \epsilon) < \hat{e}_v \leq p(1 + \epsilon)\}$. We assume $C_{-\epsilon} \subset C^*$ and hence add it to C. We then use the hybrid partition method on the set $C_{\pm\epsilon}$, meaning that we randomly partition $C_{\pm\epsilon}$ to $r = \frac{1}{\epsilon}$ sets of equal size $V_1, \ldots, V_r$. Assume we know $l_j = |V_j \cap C^*|$ for $j = 1, \ldots, r$ (these are found by exhaustive search). The algorithm begins with $C = T^* \cup C_{-\epsilon}$, and then goes over V_j, $j = 1, \ldots, r$ and adds to C the set C_j of the l_j vertices with smallest values of $\bar{c}(v) = \sum_{k<j} w(v, C_k) + \frac{r-(j-1)}{r}(\hat{e}_v - w(v, C_{-\epsilon}))$ from V_j. This step considers the distances **between** part of the vertices added to C and ensures good approximation.

Al

input G, w, p, ϵ.
output *A cluster* C, *and its value* apx.
begin
$(UC_{\min}, \text{apx}_{UC}) := UC(G, w, p, \epsilon)$.
$(BC_{\min}, \text{apx}_{BC}) := BC(G, w, p, \epsilon)$.
if $np \leq \min\{\text{apx}_{UC}, \text{apx}_{BC}\}$
  then
    return $(\emptyset, np)$.
  elseif $(\text{apx}_{UC} \leq \text{apx}_{BC})$
    then
      return $(UC_{\min}, \text{apx}_{UC})$.
    else
      return $(BC_{\min}, \text{apx}_{BC})$.
    end if
  end *Al*

Figure 2: Algorithm Al

UC

input G, w, p, ϵ.
output *A cluster* $UC_{\min}$ *and its value,* apx_{UC}.
begin
$C := \emptyset$.
$\text{apx}_{UC} := np$.
$l := |C^*|$. [*Exhaustive search.*]
for *every* $v \in V$. *[*v *is the vertex defining the cluster]*
  $C_v :=$ l *vertices* $u \in V$ *with the smallest values of* $w(u, v)$.
end for
$v^* := \arg\min w(C_v)$.
return $(C_{v^*}, w(C_{v^*}) + (n - l)p.)$
end *UC*

Figure 3: Algorithm UC

BC

input G, w, p, ϵ.
output *A cluster* $BC_{\min}$ *and its value* apx_{BC}.
begin
$\hat{D}_{C^*} := \{(1+\epsilon)^j |\ (1+\epsilon)^j \le D_{C^*} < (1+\epsilon)^{j+1}\}$. [*Exhaustive search.*]
$r := \frac{1}{\epsilon}$. [*w.l.o.g r is an integer.*]
Take a metric sample T *of* V, $|T| = \frac{8\ln(4n)}{\epsilon^4}$.
$l := |C^*|$. [*Exhaustive search.*]
$T^* := C^* \cap T$. [*Exhaustive search.*].
$\forall v \in V$, $\hat{e}_v := \frac{\hat{D}_{C^*}}{|T^*|} \sum_{u \in T^*} \frac{w(v,u)}{d_u}$.
$C_{-\epsilon} := \{v \in V |\ \hat{e}_v \le p(1-\epsilon)\}$.
if $|C_{-\epsilon}| \ge l$
  then
    $BC_{\min} := l$ *vertices of* $C_{-\epsilon}$ *with the smallest value of* $\hat{e}_v$.
    $\mathrm{apx}_{BC} := w(BC_{\min}) + (n - |BC_{\min})p$.
    return $(BC_{\min}, \mathrm{apx}_{BC})$.
  end if
$C_0 := T^* \cup C_{-\epsilon}$.
$C_{\pm\epsilon} := \{v \in V |\ p(1-\epsilon) < \hat{e}_v \le p(1+\epsilon)\}$.
Randomly partition $C_{\pm\epsilon}$ *into* r *sets* $V_1, \ldots, V_r$ *of equal size (as possible).*
$l_j := |V_j \cap C^*|$, $j = 1, \ldots, r$. [*Exhaustive search.*]
for $j = 1, \ldots, r$
  $\forall v \in V_j$, $\bar{c}(v) := \sum_{k=0}^{j-1} w(v, C_k) + \frac{r-(j-1)}{r}(\hat{e}_v - w(v, C_{-\epsilon}))$.
  $C_j := l_j$ *vertices* $v \in V_j$ *with smallest values of* $\bar{c}(v)$.
end for
$BC_{\min} := \cup_j C_j$, $\mathrm{apx}_{BC} := w(BC_{\min}) + (n-l)p$.
return $(BC_{\min}, \mathrm{apx}_{BC})$.
end *BC*

Figure 4: Algorithm BC

Let C denote the cluster returned by Algorithm *Al*. Let $P = V \setminus C$. We will analyze the following three cases separately:
Case 1: $|C^*| < \epsilon n$.
Case 2: $|P^*| < \epsilon n$.
Case 3: $|C^*| \geq \epsilon n$ and $|P^*| \geq \epsilon n$.

In our analysis we often assume that ϵ is smaller than some constant, without being specific.

Theorem 4.2 *If* $|C^*| < \epsilon n$ *then* $\text{apx} \leq (1 + 2\epsilon)\text{opt}$.

Proof: In this case $\text{opt} \geq (n - |C^*|)p \geq n(1 - \epsilon)p$, and $\text{apx} \leq np$, which yields $\frac{\text{apx}}{\text{opt}} \leq \frac{1}{1-\epsilon} \leq 1 + 2\epsilon$. ∎

For *Case 2*, $|P^*| < \epsilon n$, we use the following lemma proved in [8] under the assumption $w(C) + w(P) \leq \epsilon^2 w(V)$. Here we do not make this assumption. Note that $\frac{2w(C^*)}{|C^*|}$ is the average value of $w(v, C^*)$ for $v \in C^*$, and hence there is a vertex $z \in C^*$ for which $w(z, C^*) \leq 2\frac{w(C^*)}{|C^*|}$.

Lemma 4.3 *Assume* $|P^*| < \epsilon n$. *Let* $z \in C^*$ *such that* $w(z, C^*) \leq 2\frac{w(C^*)}{|C^*|}$. *Let* C *consist of the* $|C^*|$ *vertices* $v \in V$ *with the smallest values of* $w(v, z)$. *Then,*

$$\max\{w(C \setminus C^*, C^*) - w(C^* \setminus C, C^*), w(C^* \setminus C), w(C \setminus C^*)\} < 5\epsilon w(C^*).$$

Proof: Note that $|P^*| < \epsilon n$ is equivalent to $|P^*| < \frac{\epsilon}{1-\epsilon}|C^*|$.

Let $C \setminus C^* = \{y_1, y_2, \ldots, y_m\}$ be the set of vertices put by mistake in C, and $C^* \setminus C = \{x_1, x_2, \ldots, x_m\}$ be the set of vertices put by mistake in P. By the triangle inequality for any sets $X, Y, Z \subset V$,

$$|Z|w(X, Y) \leq |X|w(Y, Z) + |Y|w(X, Z). \tag{2}$$

Using (2) with $X = \{x\}$, $Y = C^*$ and $Z = \{z\}$ and with $X = \{x\}$, $Y = \{z\}$ and $Z = C^*$, and since $w(z, C^*) \leq 2\frac{w(C^*)}{|C^*|}$ we get for every $i = 1, \ldots, m$,

$$w(y_i, C^*) - |C^*| \cdot w(y_i, z) \leq \frac{2w(C^*)}{|C^*|}, \tag{3}$$

and

$$|C^*| \cdot w(x_i, z) - w(x_i, C^*) \leq \frac{2w(C^*)}{|C^*|}. \tag{4}$$

Note that

$$m \leq |P^*| \leq \frac{\epsilon}{1 - \epsilon}|C^*|. \tag{5}$$

By assumption $w(y_i, z) \leq w(x_i, z)$ for every $i = 1, \ldots, m$. Summing (3) and (4) over $i = 1, \ldots, m$, and by (5),

$$w(C \setminus C^*, C^*) - w(C^* \setminus C, C^*) = \sum_{i=1}^{m} [w(y_i, C^*) - w(x_i, C^*)] \leq$$

$$\sum_{i=1}^{m} \left[\Big(w(y_i, C^*) - |C^*| \cdot w(y_i, z) \Big) + \Big(|C^*| \cdot w(x_i, z) - w(x_i, C^*) \Big) \right] \leq$$

$$4 \frac{m w(C^*)}{|C^*|} \leq 4 \frac{|P^*| \cdot w(C^*)}{|C^*|} \leq \frac{4\epsilon}{1-\epsilon} w(C^*) \leq 5\epsilon w(C^*). \tag{6}$$

Next, we use (2) with $X = C^* \setminus C$, $Y = C^* \setminus C$ and $Z = C^*$, and (5), to obtain

$$w(C^* \setminus C) = \frac{1}{2} w(C^* \setminus C, C^* \setminus C) \leq w(C^* \setminus C, C^*) \frac{|C^* \setminus C|}{|C^*|} \leq 2w(C^*) \frac{|P^*|}{|C^*|} \leq 5\epsilon w(C^*). \tag{7}$$

Finally, we use (2) with $X = x_i$, $Y = y_i$ and $Z = C^*$, sum over $i = 1, \ldots, m$, and then use (6), giving

$$\begin{aligned} |C^*| \sum_{i=1}^{m} w(x_i, y_i) &\leq w(C^* \setminus C, C^*) + w(C \setminus C^*, C^*) \\ &= w(C \setminus C^*, C^*) - w(C^* \setminus C, C^*) + 2w(C^* \setminus C, C^*) \\ &\leq \left(\frac{4\epsilon}{1-\epsilon} + 2 \right) w(C^*). \end{aligned} \tag{8}$$

By the triangle inequality $w(y_i, y_j) \leq w(y_i, x_i) + w(x_i, x_j) + w(x_j, y_j)$. Summing over $i = 1, \ldots, m$ and $j = 1, \ldots, m$, and using (7), (8) and (5)

$$\begin{aligned} w(C \setminus C^*) &\leq m \sum_{i=1}^{m} w(x_i, y_i) + w(C^* \setminus C) \\ &\leq \frac{m}{|C^*|} \left(\frac{4\epsilon}{1-\epsilon} + 2 \right) w(C^*) + \frac{2\epsilon}{1-\epsilon} w(C^*) \\ &\leq \frac{4\epsilon}{1-\epsilon} \left(\frac{\epsilon}{1-\epsilon} + 1 \right) w(C^*) < 5\epsilon w(C^*). \end{aligned}$$

■

Theorem 4.4 *If* $|P^*| < \epsilon n$ *then* $\text{apx}_{UC} \leq (1 + 20\epsilon)\text{opt}$.

Proof: It is sufficient to prove the bound for the solution value generated when Algorithm AC considers a vertex $z \in C^*$ for which $w(z, C^*) \leq \frac{2w(C^*)}{|C^*|}$ and $l = |C^*|$. Therefore, we can use the bound given in Lemma 4.3.

Let $X = C^* \setminus C$ and $Y = C \setminus C^*$, then

$$\begin{aligned} \text{apx} - \text{opt} &= [w(Y, C^* \cap C) - w(X, C^* \cap C)] + [w(Y) - w(X)] \\ &= [w(Y, C^*) - w(Y, X)] - [w(X, C^*) - w(X, X)] + [w(Y) - w(X)] \\ &= [w(Y, C^*) - w(X, C^*)] + 2w(X) + w(Y) - w(Y, X) - w(X) \leq 20\epsilon w(C^*), \end{aligned}$$

where the inequality holds by Lemma 4.3. ∎

For *Case 3*, $|C^*| \geq \epsilon n$ and $|P^*| \geq \epsilon n$, we use the following lemmas:

Lemma 4.5 *Let $Y_1, \ldots, Y_n$ be independent random variables such that $0 \leq Y_i \leq b_i$ for every i. Let $Z = \sum_{i=1}^{n} Y_i$. Then,*

$$E\big[|Z - E[Z]|\big] \leq \sqrt{\frac{\sum_i b_i^2}{4}}.$$

Proof: This inequality follows directly from Lemma 2.4 in [7] since $\sigma^2(Y_i) \leq \frac{b_i^2}{4}$. ∎

Consider a given value t, and a set $U \subset V$. Let $T = \{u_1, \ldots, u_t\}$ be a metric sample of U of size t. For every vertex $v \in V$ define

$$e_v = \frac{D_U}{t} \sum_{u \in T} \frac{w(v, u)}{d_u}.$$

Lemma 4.6 *[7]*

$$Pr\big[|w(v, U) - e_v| \leq \epsilon w(v, U)\big] \geq 1 - 2e^{-t\epsilon^2/8},$$

and

$$E\big[|w(v, U) - e_v|\big] \leq \frac{2}{\sqrt{t}} w(v, U).$$

Lemma 4.7 *Let $T = \{u_1, \ldots, u_t\}$ be a metric sample of V where $t \geq \frac{2}{\epsilon^4}$, and let $C \subset V$ where $|C| \geq \epsilon n$. Then $T \cap C$ is a metric sample of C, and $Pr\big[|T \cap C| \geq t\epsilon^2\big] \geq 1 - \epsilon$.*

Proof: Clearly $C \cap T$ is a metric sample of size $|T \cap C|$ of C. It is sufficient to prove the inequality for the boundary case $|C| = \epsilon n$ and $|P| = |V \setminus C| = (1 - \epsilon)n$. By (2) with $X = Y = P$ and $Z = C$, $\epsilon n \cdot w(P, P) \leq 2(1 - \epsilon)n \cdot w(P, C)$. Therefore,

$$D_P = w(P, P) + w(P, C) \leq w(P, C) \left(1 + \frac{2(1 - \epsilon)}{\epsilon}\right).$$

Also, $D_C \geq w(C, P)$, and by the metric sample definition, for $i = 1, \ldots, t$

$$Pr[u_i \in C] \geq \frac{D_C}{D_C + D_P} \geq \frac{\epsilon}{\epsilon + \epsilon + 2(1 - \epsilon)} = \frac{\epsilon}{2}.$$

The random variable $|C \cap T|$ stochastically dominates the binomial random variable $X \sim B(t, \frac{\epsilon}{2})$, and by the Central Limit Theorem, for $t \geq \frac{1}{\epsilon^4}$ and $\epsilon \leq \frac{1}{5}$,

$$\begin{aligned} Pr[X \geq 2\epsilon E[X]] &= 1 - Pr[X < 2\epsilon E[X]] \\ &\geq 1 - \Phi\left(\frac{(2\epsilon - 1)\frac{\epsilon}{2}t}{\sqrt{(1-\frac{\epsilon}{2})\frac{\epsilon}{2}t)}}\right) \\ &\geq 1 - \Phi\left(-\epsilon\sqrt{\frac{\epsilon}{2}t}\right) \\ &\geq 1 - \Phi\left(-\sqrt{\frac{1}{\epsilon}}\right) \geq 1 - \epsilon. \end{aligned}$$

■

Recall the definitions of $C_{-\epsilon}$ and $C_{\pm\epsilon}$ in Algorithm *BC*. We also define $C_{+\epsilon} := \{v \in V | \hat{e}_v \leq p(1+\epsilon)\}$. Note that $C_{-\epsilon}$, $C_{\pm\epsilon}$ and $C_{+\epsilon}$ are random variables defined by the metric sample T.

Remark 4.8 To simplify the presentation, we assume in the following that $D_{C^*} = \hat{D}_{C^*}$, implying $e_v = \hat{e}_v$ for every $v \in V$. Since $\frac{D_{C^*}}{\hat{D}_{C^*}} < 1 + \epsilon$, then also $e_v = \frac{D_{C^*}}{\hat{D}_{C^*}}\hat{e}_v < (1+\epsilon)\hat{e}_v$, and the real value of the solution is at most $1 + \epsilon$ times the value of the solution under this assumption.

Lemma 4.9 *Consider Algorithm BC.*

1. $Pr\left[\left(|T \cap C^*| \geq \frac{8\ln(4n)}{\epsilon^2}\right) \wedge (w(v, C^*) \leq 2p \ \forall v \in C_{+\epsilon})\right] \geq \frac{3}{4}(1-\epsilon)$.
2. $E\big[|P^* \cap C_{-\epsilon}|\big] \leq 1$ *and* $E[w(C, P^* \cap C_{-\epsilon})] \leq 4\epsilon\text{opt}$.
3. $E\big[|C^* \setminus C_{+\epsilon}|\big] \leq 1$ *and* $E[w(C^* \setminus C_{+\epsilon}, C^*)] \leq \epsilon\text{opt}$.

Proof: By Lemma 4.7, $T \cap C^*$ is a metric sample of C^* and

$$Pr\big[|T \cap C^*| \geq \epsilon^2 |T|\big] = Pr\left[|T \cap C^*| \geq \frac{8\ln(4n)}{\epsilon^2}\right] \geq 1 - \epsilon. \tag{9}$$

Assume $|T \cap C^*| \geq \frac{8\ln(4n)}{\epsilon^2}$, $v_1, \ldots, v_q$. Note that q and $v_1, \ldots, v_q$ are random variables resulting from the choice of the metric sample. Then, for $i = 1, \ldots, q$

$$\begin{aligned} Pr[w(v_i, C^*) \geq 2p] &\leq Pr\left[w(v_i, C^*) \geq \frac{1+\epsilon}{1-\epsilon}p\right] \\ &= Pr\big[(1-\epsilon)w(v_i, C^*) \geq (1+\epsilon)p\big] \\ &\leq Pr\big[(1-\epsilon)w(v_i, C^*) \geq e_v\big] \\ &\leq 2e^{-\epsilon^2|T\cap C^*|/8} \\ &\leq 2e^{-\ln(4n)} = \frac{1}{2n}, \end{aligned}$$

where the second inequality holds since, under the assumption $\hat{e}_v = e_v$, for $v \in C_{+\epsilon}$, $e_v \leq (1+\epsilon)p$, and the third inequality holds due to Lemma 4.6. Hence for large n,

$$Pr[w(v, C^*) \leq 2p \ \forall v \in C_{+\epsilon}] \geq \left(1 - \frac{1}{2n}\right)^n \geq \frac{3}{4}.$$

This inequality with (9) concludes the proof of part 1.

Suppose that $P^* = \{u_1, \ldots, u_r$. Then for $v = u_i$, $i = 1, \ldots, r$, $w(v, C^*) \geq p$ since otherwise a better solution is created by adding v to C^*. By Lemma 4.7 $T \cap C^*$ is a metric sample of C^* and $Pr\big[|T \cap C^*| \geq \epsilon^2 |T|\big] = Pr\left[|T \cap C^*| \geq \frac{8\ln(4n)}{\epsilon^2}\right] \geq 1 - \epsilon$, and hence by Lemma 4.6,

$$\begin{aligned} Pr[e_v \leq (1-\epsilon)p] &\leq Pr[e_v \leq (1-\epsilon)w(v, C^*)] \\ &= Pr[w(v, C^*) - e_v \geq \epsilon w(v, C^*)] \\ &\leq 2e^{-\epsilon^2 |T \cap C^*|/8} \leq \frac{1}{2n}, \end{aligned}$$

and therefore

$$E\big[|P^* \cap C_{-\epsilon}|\big] \leq \frac{1}{2n} n \leq 1. \tag{10}$$

Consider $y \in C$ and $v \in P^* \cap C_{-\epsilon}$. Using (2) with $X = \{y\}$, $Y = \{v\}$ and $Z = C^*$, and using the first part of the lemma, then with probability $\frac{3}{4}(1-\epsilon)$,

$$w(v, y) \leq \frac{w(v, C^*) + w(y, C^*)}{|C^*|} \leq \frac{4p}{|C^*|}. \tag{11}$$

The proof of part 2 is concluded by

$$E[w(C, P^* \cap C_{-\epsilon})] \leq E\big[|C|\big] E\big[|P^* \cap C_{-\epsilon}|\big] \frac{4p}{|C^*|} \leq 4p \leq \frac{4}{\frac{1}{\epsilon^2}} \text{opt} \leq 4\epsilon^2 \text{opt}, \tag{12}$$

where the first inequality follows from (11) and $|C| = |C^*|$, the second from (10), and the third from (1) and $\text{opt} \geq p|P^*|$. This concludes the proof of part 2.

Since obviously, $w(v, C^*) \leq p$ for $v \in C^*$, and by Lemma 4.6

$$\begin{aligned} Pr[e_v \geq (1+\epsilon)p] &\leq Pr[e_v \geq (1+\epsilon)w(v, C^*)] \\ &= Pr[e_v - w(v, C^*) \geq \epsilon w(v, C^*)] \\ &\leq 2e^{-t\epsilon^2/8} \leq \frac{1}{2n}, \end{aligned}$$

where $t = |T|$, and

$$E\big[|C^* \setminus C_{+\epsilon}|\big] \leq \frac{1}{2n} n \leq 1.$$

Finally, from $w(v, C^*) \leq p$ for $v \in C^*$, and $|P^*| \geq \frac{1}{\epsilon^2}$, it follows that $E[w(C^* \setminus C_{+\epsilon}, C^*)] \leq p \leq \epsilon^2 \text{opt}$. ∎

In the following we assume that $C_{-\epsilon} \subseteq C^*$ and that $C^* \subseteq C_{+\epsilon}$. It follows from the third part of Lemma 4.9 that with probability $\frac{3}{4}(1-\epsilon)$ the expected weight of the errors due to this assumption is $O(\epsilon \text{opt})$.

The following lemma is based on the deterministic analysis in [7]. For $j = 1, \ldots, r$, let $C_j^* = C^* \cap V_j$, and let I_j denote the following *left part* of the hybrid partitioning:

$$I_j = (\bigcup_{k=0}^{j} C_k) \cup (\bigcup_{k=j+1}^{r} C_k^*).$$

Under the assumptions $C_{-\epsilon} \subseteq C^*$ and $C^* \subseteq C_{+\epsilon}$, $I_0 = C^*$ and $I_r = C$. For $j = 1, \ldots, r$, consider the points that are classified differently in I_j and I_{j-1}. Let $X_j := C_j^* \setminus C_j = \{x_1, \ldots, x_m\}$ and $Y_j := C_j \setminus C_j^* = \{y_1, \ldots, y_m\}$.

Lemma 4.10 *For $j = 1, \ldots, r$*

$$\begin{aligned} E\big[w(I_j) - w(I_{j-1})\big] \quad \leq \quad & sum_{u \in X_j \cup Y_j} E\Big[\Big|w(u, \bigcup_{k=j}^{r} C_k^*) - \frac{r-j+1}{r} w(u, C^* \cap C_{\pm\epsilon})\Big| \\ & + |w(u, C^*) - e_u|\Big] + E[w(Y_j, X_j)]. \end{aligned}$$

Proof:

$$\begin{aligned} w(I_j) - w(I_{j-1}) &= w(Y_j, I_{j-1} \setminus X_j) + w(Y_j) - [w(X_j, I_{j-1} \setminus X_j) + w(X_j)] \\ &= w(Y_j, I_{j-1}) - w(Y_j, X_j) + w(Y_j) - [w(X_j, I_{j-1}) - w(X_j, X_j) + w(X_j)] \\ &= w(Y_j, I_{j-1}) - w(X_j, I_{j-1}) + w(Y_j) + w(X_j) - w(Y_j, X_j) \\ &\leq \sum_{i=1}^{|Y_j|} [w(y_i, I_{j-1}) - w(x_i, I_{j-1})] + w(Y_j, X_j) \\ &\leq \sum_{i=1}^{|Y_j|} [w(y_i, I_{j-1}) - \bar{c}(y_i) + \bar{c}(x_i) - w(x_i, I_{j-1})] + w(Y_j, X_j) \\ &\leq \sum_{u \in X_j} |w(u, I_{j-1}) - \bar{c}(u)| + \sum_{u \in Y_j} |w(u, I_{j-1}) - \bar{c}(u)| + w(Y_j, X_j) \\ &= \sum_{u \in X_j \cup Y_j} |w(u, I_{j-1}) - \bar{c}(u)| + w(Y_j, X_j). \end{aligned} \tag{13}$$

The first inequality is due to (2) with $X = Y = Y_j$, $Z = X_j$ and $|X_j| = |Y_j|$, giving $w(Y_j) \leq w(X_j, Y_j)$. Similarly, $w(X_j) \leq w(X_j, Y_j)$. The second inequality holds since $\bar{c}(y_i) \leq \bar{c}(x_i)$.

Under the assumption $\hat{e}_v = e_v$, for $u \in V_j$,

$$\bar{c}(u) = \sum_{k=0}^{j-1} w(u, C_k) + \frac{r-(j-1)}{r}[e_u - w(u, C_{-\epsilon})],$$

and therefore,

$$\begin{aligned}
|w(u, I_{j-1}) - \bar{c}(u)| &= \Big|w(u, \bigcup_{k=j}^{r} C_k^*) - \frac{r-j+1}{r}[e_u - w(u, C_{-\epsilon})]\Big| \\
&= \Big|w(u, \bigcup_{k=j}^{r} C_k^*) - \frac{r-j+1}{r}\Big[e_u + w(u, C^* \cap C_{\pm\epsilon}) - w(u, C^*)\Big]\Big| \\
&\leq \Big|w(u, \bigcup_{k=j}^{r} C_k^*) - \frac{r-j+1}{r} w(u, C^* \cap C_{\pm\epsilon})\Big| + |w(u, C^*) - e_u|. \qquad (14)
\end{aligned}$$

Substituting (14) into (13) and taking the expectation on both sides completes the proof of the lemma. ∎

Theorem 4.11 *If $|C^*| \geq \epsilon n$ and $|P^*| \geq \epsilon n$, then with probability of at least $\frac{9}{16}(1-\epsilon)$, $\text{apx}_{BC} \leq (1 + 32\epsilon)\text{opt}$.*

Proof: In the following we assume for every $v \in C_{+\epsilon}$

$$w(v, C^*) \leq 2p, \qquad (15)$$

and $|T \cap C^*| \geq \frac{8\ln(4n)}{\epsilon^2}$. By the first part of Lemma 4.9, these assumptions hold with probability of at least $\frac{1}{2}(1-\epsilon)$.

Fix $j \in \{1, \ldots, r\}$ and $u \in X_j \cup Y_j$ and let $Z_u = w(u, \bigcup_{k=j}^{r} C_k^*) = \sum_{k=j}^{r} w(u, C_k^*)$. Since $C_{\pm\epsilon}$ is randomly partitioned into $V_1, \ldots, V_r$, $E[Z_u] = \frac{r-j+1}{r} w(u, C^* \cap C_{\pm\epsilon})$ and $Z_u = \sum_{s \in C^* \cap C_{\pm\epsilon}} w(u, s) A_s$, where $\{A_s\}$ are 0/1 i.i.d. random variables with $Pr[A_s = 1] = \frac{r-j+1}{r}$.

We use (2) with $X = \{u\}$, $Y = \{s\}$ and $Z = C^*$, (15), and the fact that $w(s, C^*) \leq p$ for $s \in C^*$, to obtain for every $u \in C_{\pm\epsilon}$ and $s \in C^*$

$$w(u, s) \leq \frac{w(u, C^*) + w(s, C^*)}{|C^*|} \leq \frac{3p}{|C^*|}. \qquad (16)$$

We use Lemma 4.5 for $Z_u = \sum_{s \in C^* \cap C_{\pm\epsilon}} w(u, s) A_s := \sum_{s \in C^* \cap C_{\pm\epsilon}} Q_s$, where Q_s, for every $s \in C^* \cap C_{\pm\epsilon}$, is nonnegative and bounded by $b_s = \frac{3p}{|C^*|}$, to obtain

$$E\big[|Z_u - E[Z_u]|\big] \leq \sqrt{\frac{\sum_{s \in C^* \cap C_{\pm\epsilon}} 9p^2}{4|C^*|^2}} \leq \frac{3p}{2\sqrt{|C^*|}},$$

and therefore, for every $u \in X_j \cup Y_j$,

$$E\Big[\Big|w(u, \bigcup_{k=j}^{r} C_k^*) - \frac{r-j+1}{r} w(u, C^* \cap C_{\pm\epsilon})\Big|\Big] = E\big[|Z_u - E[Z_u]|\big] \leq \frac{3p}{2\sqrt{|C^*|}}. \qquad (17)$$

By the second part of Lemma 4.6, (15), and the assumption $|T \cap C^*| \geq \frac{8\ln(4n)}{\epsilon^2}$, for every $u \in X_j \cup Y_j$,

$$E\big[|w(u,C^*) - e_u|\big] \leq \frac{2w(u,C^*)}{\sqrt{|T \cap C^*|}} \leq \frac{\epsilon w(u,C^*)}{\sqrt{2\ln(4n)}} \leq \frac{\sqrt{2}\epsilon p}{\sqrt{\ln(4n)}}. \tag{18}$$

Substituting (17) and (18) in Lemma 4.10,

$$\begin{aligned} E[w(I_j) - w(I_{j-1})] &\leq E\left[w(Y_j,X_j) + \sum_{u \in X_j \cup Y_j}\left(\frac{3p}{2\sqrt{|C^*|}} + \frac{\epsilon p}{\sqrt{\ln(4n)}}\right)\right] \\ &= E(w(Y_j,X_j)) + 2E|Y_j|\left[\frac{3p}{2\sqrt{|C^*|}} + \frac{\epsilon p}{\sqrt{\ln(4n)}}\right] \\ &\leq E(w(Y_j,X_j)) + 4\epsilon^2|P^*|p. \end{aligned} \tag{19}$$

The second inequality holds since by (1) $|C^*| \geq \frac{1}{\epsilon^2}$ and $|P^*| \geq \frac{1}{\epsilon^2}$, and by assumption $E\big[|Y_j|\big] \leq \epsilon|P^*|$. Let $x \in X_j$ and $y \in Y_j$.

Recall that we randomly partitioned P^* into $r = \frac{1}{\epsilon}$ subsets $V_1 \ldots, V_r$ giving each vertex an equal probability to be in any of the subsets. Therefore, $E[|V_j|] = \epsilon|P^*|$, and since $Y_j \subseteq V_j$ $E[|Y_j|^2] \leq E[|V_j|^2] = (1-2\epsilon)\epsilon|P^*|$. Summing (16) over all $x \in X_j$ and $y \in Y_j$ gives

$$E[w(X_j,Y_j)] \leq \frac{E\big[|Y_j| \cdot |X_j|\big]3p}{|C^*|} = \frac{E\big[|Y_j|^2\big]3p}{|C^*|} \leq 3\epsilon|P^*|p \leq 3\epsilon^2\text{opt}. \tag{20}$$

Substituting (20) into (19) and noting that $\text{opt} \geq |P^*|p$,

$$E[w(I_j) - w(I_{j-1})] \leq 8\epsilon^2\text{opt}. \tag{21}$$

Summing (21) over $j = 1, \ldots, r = \frac{1}{\epsilon}$ gives,

$$E[w(C) - w(C^*)] = E[w(I_r) - w(I_0)] \leq 8\epsilon\text{opt}. \tag{22}$$

By Markov's inequality with probability $\frac{3}{4}(1-\epsilon)$,

$$Pr\left[w(C) - w(C^*) \leq 32\epsilon\text{opt})\right] \geq 1 - \frac{E[w(C) - w(C^*)]}{32\epsilon\text{opt}} \geq 1 - \frac{8}{32} = \frac{3}{4}.$$

■

Theorem 4.12 *There is a polynomial time approximation algorithm for PMS.*

Proof: By running Algorithm *Al* $O\left(\log \frac{1}{1-\epsilon}\right)$ times, we obtain the approximation ratio as follows from Theorems 4.2, 4.4 and 4.11. ■

Remark 4.13 Algorithm Al can be generalized to give a randomized approximation scheme for metric w with non-uniform penalties, assuming a constant bound on the maximal to minimal penalty ratio. For simplicity, assume that the penalties p_v are integers in $\{1, \ldots, p\}$, where p is a given constant. The running time of the algorithm depends on p and ϵ. We briefly outline the changes needed to accommodate this generalization.

A similar proof as in Theorem 4.2 shows that if $|C^*| < \epsilon n$, then the solution $C = \emptyset$ is a good approximation. In this case, $\text{opt} \geq n(1-\epsilon)$ whereas $\text{apx} \leq \text{opt} + \epsilon np$, so that $\frac{\text{apx}}{\text{opt}} \leq 1 + \epsilon(1+\epsilon)p$.

Define $D_i = \{v \in V | p_v = i\}$. The adaptation of Algorithm UC is based on finding by exhaustive search the values $l^i = |C^* \cap D_i|$, so that C includes the optimal number of vertices from each penalty value. This property allows us to assume that for every vertex of penalty i put by mistake in C, there is a vertex of penalty value i put by mistake in P, i.e. the sets X and Y can be paired in such a way that the penalty of the vertices within each pair is identical. The metric sample from C^* taken in BC to estimate the distance to C^* is as efficient in this case as it was in the uniform case. However, we do require that the values $l^i_j = |C^* \cap D_i \cap V_j|$ will be found by exhaustive search, so that we may assume that for every vertex of penalty i put by mistake in C_j, there is a vertex of penalty value i put by mistake in P_j, i.e., the sets of errors X_j and Y_j can be paired so the penalty of the vertices within each pair is identical.

4.1 2-approximation for metric kPMS

In this section we generalize [12], which offers a 2-approximation for the metric K-MIN-SUM problem for a fixed k. We suggest the following algorithm:

- Let $G' = (V', E')$ where $V' = V \cup \{z\}$, $E' = E \cup \{(v, z) | v \in V\}$, and $w(v, z) = p_v$ for every $v \in V$.
- For every set of sizes $l_1, \ldots, l_{k+1}$ satisfying $\sum_{i=1}^{k+1} l_i = n$, and k distinct vertices $\{v_i\}_{i=1}^{k} \subset V$, compute a partition of V into $k+1$ disjoint sets $\{S_1, \ldots, S_k, S_{k+1}\}$ of sizes $l_1, \ldots, l_{k+1}$, such that the cost $\sum_{i=1}^{k} l_i w(v_i, S_i) + \sum_{v \in S_{k+1}} w(z, v)$ is minimized.
- Return the partition of minimum cost $\{S^*_1, \ldots, S^*_{k+1}\}$, where $S^*_1, \ldots, S^*_k$ are the clusters, and S^*_{k+1} is the set of unclustered vertices.

Let $\text{apx} = \sum_{i=1}^{k} w(S^*_i) + \sum_{v \in S^*_{k+1}} p_v$. Denote the cost of the optimal solution by opt.

Claim 4.14 $\text{apx} \leq 2\text{opt}$.

Proof: The proof is identical to that of Theorem 3.1 in [12], the only change is that the sum of penalties over S^*_{k+1} is added to apx and the corresponding sum is added to opt. ∎

References

[1] A. Aboud and Y. Rabani, "Correlation clustering with penalties," manuscript, 2006.

[2] R.O. Anstee, "A polynomial algorithm for b-matchings: an alternative approach," *Information Processing Letters* **24** (1987), 153-157.

[3] M. Charikar, S. Khuller, D.M. Mount, and G. Narasimhan, "Algorithms for facility location problems with outliers," *SODA* (2001), 642-651.

[4] K. Chen, "A constant factor approximation algorithm for k-median clustering with outliers," *Proceedings of SODA'08* (2008), 826-835.

[5] I. Dinur and S. Safra, "On the hardness of approximating minimum vertex-cover," *Annals of Mathematics* **162** (2005), 439-485.

[6] W. Fernandez de la Vega, and C. Kenyon, "A randomized approximation scheme for metric MAX-CUT," *J. Comput. Science* **63** (2001), 531-541.

[7] W. Fernandez de la Vega, M. Karpinski and C. Kenyon, "Approximation schemes for metric bisection and partitioning," *SODA'04: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, (2004) 506-511.

[8] W. Fernandez de la Vega, M. Karpinski, C. Kenyon and Y. Rabani, "Approximation schemes for clustering problems," *Proceedings of the 35th ACM STOC* (2003) 50-58.

[9] U. Feige, G. Kortsarz and D. Peleg, "The dense k-subgraph problem," *Algorithmica* **29** (2001), 410-421.

[10] O. Goldreich, S. Goldwasser and D. Ron, "Property testing and its connection to learning and approximation," *Journal of the ACM* **45** (1998), 653-750.

[11] A. Gupta, R. Krishnaswamy, A. Kumar, and D. Segev, "Scheduling with outliers," manuscript, 2008.

[12] N. Guttmann-Beck and R. Hassin, "Approximation algorithms for min-sum p-clustering," *Discrete Applied Mathematics* **89** (1998), 125-142.

[13] M.R. Garey and D.S. Johnson "Computers and Intractability," *Freeman* (1979).

[14] N. Garg, V. V. Vazirani, and M. Yannakakis, "Multiway cuts in directed and node weighted graphs," *Proc. 21st Int. Colloquium on Automata, Languages and Programming, Lecture Notes in Comput. Sci. 820, Springer-Verlag*, (1994),487-498.

[15] D. S. Hochbaum, "Solving integer programs over monotone inequalities in three variables: a framework for half integrality and good approximation," *European Journal of Operational Research* **140** (2002), 291-321.

[16] R. Hassin and A. Levin, "The minimum generalized vertex cover problem," *ACM Transactions on Algorithms* **2** (2006), 66-78.

[17] R. Hassin, S. Rubinstein and A. Tamir, "Approximation algorithm for maximum dispersion," *Operations Research Letters* **21** (1997),133-137.

[18] P. Indyk, "A sublinear time approximation scheme for clustering in metric spaces," *Proceedings of the 40th Symposium on Foundations of Computer Science* (1999), 154-159.

[19] S. Khot and O. Regev, "Vertex Cover Might be Hard to Approximate to within $2-\epsilon$," *IEEE Transactions on Information Theory* **50** (2004), 2031-2037.

[20] E. Lawler, *Combinatorial Optimization, Networks and Matroids*, Dover publications, (1976).

[21] S. Sahni, T. Gonzalez "P-complete approximation problems," *Journal of the ACM* **23** (1976), 555-566.

[22] G. Xu, J. Xu, "An LP rounding algorithm for approximating uncapacitated facility location problem with penalties [rapid communication]," *Information Processing Letters* **94** (2005), 119-123.