



An improved approximation algorithm for the metric maximum clustering problem with given cluster sizes

Refael Hassin*, Shlomi Rubinstein

Department of Statistics and Operations Research, School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel

Received 28 June 2005; received in revised form 30 November 2005; accepted 5 December 2005

Available online 5 January 2006

Communicated by K. Iwama

Abstract

The input to the METRIC MAXIMUM CLUSTERING PROBLEM WITH GIVEN CLUSTER SIZES consists of a complete graph $G = (V, E)$ with edge weights satisfying the triangle inequality, and integers c_1, \dots, c_p . The goal is to find a partition of V into disjoint clusters of sizes c_1, \dots, c_p , maximizing the sum of weights of edges whose two ends belong to the same cluster. We describe an approximation algorithms for this problem with performance guarantee that approaches 0.5 when the cluster sizes are large.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Approximation algorithms; Maximum weight clustering

1. Introduction

In this paper we approximate the METRIC MAXIMUM CLUSTERING PROBLEM WITH GIVEN CLUSTER SIZES. The input for the problem consists of a complete graph $G = (E, V)$, $V = \{1, \dots, n\}$, with nonnegative edge weights $w(i, j)$, $(i, j) \in E$, that satisfy the triangle inequality, and cluster sizes c_1, \dots, c_p , where $\sum_{i=1}^p c_i \leq n$. The problem is to partition V into sets of the given sizes, so that the total weight of edges inside the clusters is maximized.

In [3] we gave a approximation algorithm whose error ratio is bounded by $\frac{1}{2\sqrt{2}} \approx 0.353$. In [4] we improved this result for the case in which cluster sizes are large.

In particular, when the minimum cluster size increases, the performance guarantee increases asymptotically to 0.375. Special cases, with and without the metric assumption, were considered in [1,2,5,6].

In this paper we present a randomized $(\frac{1}{2} - \frac{3}{k})$ -approximation algorithm for the problem, where k is the size of the smallest cluster. Thus, for large clusters the bound is asymptotically $\frac{1}{2}$, as the best known asymptotic bound for the same problem but with identical cluster sizes [2].

A p -matching is a set of p vertex-disjoint edges in a graph. A greedy p -matching is obtained by sorting the edges in non-increasing order of their weights, and then scanning the list and selecting edges as long as they are vertex-disjoint to the previously selected edges, and their number does not exceed p . In a graph with k vertices, a perfect matching has $k/2$ edges if k is even, and $(k-1)/2$ edges if k is odd. For a matching M we de-

* Corresponding author.

E-mail addresses: hassin@post.tau.ac.il (R. Hassin),
shlomiru@post.tau.ac.il (S. Rubinstein).

note by $V(M)$ its vertex set and by $E(V(M))$ the edges in the subgraph induced by $V(M)$. For a set of edges F we denote by $w(F)$ the sum of weights of the edges in F . We will use the following property on a perfect matching in a metric:

Lemma 1. Consider a complete graph $G' = (V', E')$ with k vertices, and a metric $w_e, e \in E'$. Let M' be a perfect matching on G' . Then $w(M') \leq \frac{2}{k}w(E')$.

Proof. Suppose first that k is even. By the triangle inequality, $w(v, a) + w(v, b) \geq w(e)$ for every $e = \{a, b\} \in M'$ and $v \in G' \setminus e$. Summing over all such v and e gives $2[w(E') - w(M')] \geq (k - 2)w(M')$, or $w(M') \leq \frac{2}{k}w(E')$.

Similarly, if k is odd then we sum twice the edges incident with the vertex u that does not belong to M' to obtain $2[w(E(V(M'))) - w(M')] + 2 \sum_{v \in V(M')} w(u, v) \geq (k - 1)w(M')$ giving $w(M') \leq \frac{2}{k+1}w(E')$. \square

For $i \leq \lfloor n/2 \rfloor$, we denote by M_i a maximum i -matching. Thus, $|V(M_i)| = 2|M_i|$.

Lemma 2. It is possible to choose the maximum matchings $\{M_i\}$ so that $V(M_1) \subset V(M_2) \subset \dots \subset V(M_{\lfloor n/2 \rfloor})$.

Proof. The proof is by induction on i . Suppose that there exists $v \in M_i \setminus M_{i+1}$, then $M_i \cup M_{i+1}$ contains an alternating path P with end vertex v . In particular, $|P \cap M_i| \geq |P \cap M_{i+1}|$. If $|P \cap M_i| = |P \cap M_{i+1}|$ then by the optimality of M_i and M_{i+1} these two sets must have identical weight. We can swap the edges of $P \cap M_{i+1}$ by those of $P \cap M_i$ and obtain a new maximum $(i + 1)$ -matching that uses v . If $|P \cap M_i| = |P \cap M_{i+1}| + 1$ then there must be another alternating path P' such that $|P' \cap M_i| + 1 = |P' \cap M_{i+1}|$. Again, by the optimality of M_i and M_{i+1} the weight of $(P \cap M_i) \cup (P' \cap M_i)$ and $(P \cap M_{i+1}) \cup (P' \cap M_{i+1})$ must be the same. We can swap the edges in both paths to obtain a new maximum $(i + 1)$ -matching that uses v . Repeating this step, we end up with a maximum $(i + 1)$ -matching whose vertex set contains $V(M_i)$. \square

2. The algorithm

Let the cluster sizes be ordered so that $c_1 \geq \dots \geq c_p$. Denote $q = \lfloor c_1/2 \rfloor$.

Algorithm *Metric* is presented in Fig. 2. The algorithm partitions V into layers, L_1, \dots, L_{q+1} . The last layer, L_{q+1} , consists of a single vertex for each of the odd-sized clusters. Each of the first q layers consists of pairs of vertices, one pair for each active cluster.

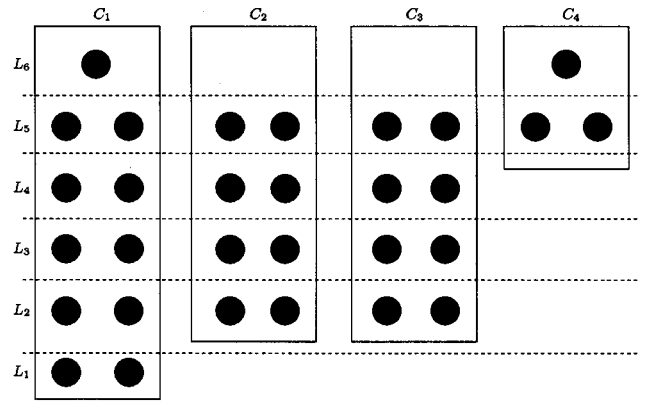


Fig. 1. The layer structure: An example with $n = 30$, $(c_1, \dots, c_4) = (11, 8, 8, 3)$, $(r_1, \dots, r_5) = (1, 3, 3, 3, 4)$, $(m_1, \dots, m_5) = (1, 4, 7, 10, 14)$, $q = 5$, and $S_{\text{odd}} = \{1, 4\}$.

A cluster is active if the number of yet unassigned vertices (rounded down to an even integer) for this cluster is maximal among all clusters.¹ The number of active clusters grows during the general (that is, excluding the last) steps of the algorithm, till they all become active. During the j th iteration, the algorithm computes a maximum matching of size m_j . The increase, $r_j = m_j - m_{j-1}$, in the size of this matching, is equal to the number of active clusters. The newly used vertices, $L_j = V(M_{m_j}) \setminus V(M_{m_{j-1}})$, are distributed among the active clusters, two to each cluster. This distribution is done randomly. When the algorithm terminates, all clusters reach their sizes. Fig. 1 illustrates the layer structure.

Let $W_j = w(M_{m_j})$ denote the weight of the maximum m_j -matching.

The main step of the algorithm randomly distributes pairs of vertices from the new layer among the active clusters. The next lemma gives a lower bound on the expected weight added to the solution by this allocation.

Lemma 3. Consider $v \in L_{j+1}$, $j \in \{1, \dots, q - 1\}$. Let α_v be the expected weight of the edges connecting v and vertices from $V(M_{m_j})$ in the cluster to which v is added. Then, $\alpha_v \geq W_j/r_{j+1}$.

Proof. Consider an edge $\{v, u\}$ where $u \in V(M_{m_j})$. The probability that this edge contributes its weight to α_v is $1/r_{j+1}$ since there are r_{j+1} active clusters and v is inserted to each of them with equal probability. Therefore,

$$\alpha_v = \frac{1}{r_{j+1}} \sum_{u \in V(M_{m_j})} w(v, u). \quad (1)$$

¹ C_1 is always active and therefore the number of such layers is q .

Metric

input

1. A complete undirected graph $G = (V, E)$ with a metric $w(e)$, $e \in E$.
2. Integers $c_1 \geq \dots \geq c_p$ such that $\sum_i c_i \leq |V|$.

returns

Clusters C_1, \dots, C_p such that $|C_i| = c_i$.

begin

$q := \lfloor c_1/2 \rfloor$.

$C_i := \emptyset$, $i = 1, \dots, p$.

$a_i := 2\lfloor c_i/2 \rfloor$, $i = 1, \dots, p$.

$S_{\text{odd}} := \{i : a_i = c_i - 1\}$.

$m_0 := 0$.

for every $j = 1, \dots, q$

$r_j := \max\{i : a_i = a_1\}$. [Clusters C_1, \dots, C_{r_j} are active.]

$m_j := m_{j-1} + r_j$.

Compute a maximum m_j -matching M_{m_j} such that $V(M_{m_{j-1}}) \subset V(M_{m_j})$.

$L_j := V(M_{m_j}) \setminus V(M_{m_{j-1}})$. [L_j is a layer.]

Randomly partition L_j into pairs and add one pair to each active cluster.

$a_i := a_i - 2$, $i = 1, \dots, r_j$.

Randomly select a yet unused vertex to each C_i , $i \in S_{\text{odd}}$.

return C_1, \dots, C_p .

end Metric

Fig. 2. Algorithm Metric.

Consider an edge $e = \{a, b\} \in M_{m_j}$. By the triangle inequality, $w(v, a) + w(v, b) \geq w(e)$. Summation over $e \in M_{m_j}$ gives

$$\sum_{u \in V(M_{m_j})} w(v, u) \geq W_j.$$

With (1), this inequality proves the claim. \square

Consider an optimal solution OPT with clusters O_1, \dots, O_p of sizes c_1, \dots, c_p , respectively. For $i = 1, \dots, p$ and $j = 1, \dots, \lfloor c_i/2 \rfloor$, let $G_{i,j}$ be a greedy j -matching on O_i such that $G_{i,1} \subset \dots \subset G_{i,\lfloor c_i/2 \rfloor}$. Let $w(G_{i,j})$ denote the weight of $G_{i,j}$, and let $e_{i,j} = G_{i,j} \setminus G_{i,j-1}$, be the j th edge added to the greedy matching in O_i .

Lemma 4. Consider a vertex $v \in O_i \setminus V(G_{i,j})$. Let β_v denote the weight of the edges connecting v and $V(G_{i,j})$. Then, $\beta_v \leq 2w(G_{i,j})$.

Proof. Consider an edge $e = \{a, b\} \in G_{i,j}$. Since $G_{i,j}$ is a greedy matching, $w(a, b) \geq w(v, a)$, $w(v, b)$, and thus $w(v, a) + w(v, b) \leq 2w(a, b)$. Therefore,

$$\begin{aligned} \beta_v &= \sum_{f \in V(G_{i,j})} w(v, f) \leq 2 \sum_{e \in G_{i,j}} w(e) \\ &= 2w(G_{i,j}). \quad \square \end{aligned}$$

Theorem 1. Let $k = c_p > 6$. Algorithm Metric returns a $(\frac{1}{2} - \frac{3}{k})$ -approximation.

Proof. Let apx denote the expected weight of the solution returned by the algorithm. Then,

$$apx \geq \sum_{j=1}^{q-1} \sum_{v \in L_{j+1}} \alpha_v \geq \sum_{j=1}^{q-1} 2r_{j+1} \frac{W_j}{r_{j+1}} = \sum_{j=1}^{q-1} 2W_j. \quad (2)$$

The first inequality follows since the summation is over a subset of the edges of the approximate solution. The second inequality follows from Lemma 3 and since $|L_{j+1}| = 2r_{j+1}$.

Let $GR = \bigcup_{i=1}^p G_{i,\lfloor c_i/2 \rfloor}$, and for $i \in S_{\text{odd}}$ let $\{v_i\} = V(O_i) \setminus V(G_{i,\lfloor c_i/2 \rfloor})$ be the vertex left out by the greedy algorithm in O_i . Denote by opt the value of an optimal solution. Then,

$$\begin{aligned} opt &= \sum_{i=1}^p \sum_{j=1}^{\lfloor c_i/2 \rfloor - 1} \sum_{v \in e_{i,j+1}} \beta_v + \sum_{i \in S_{\text{odd}}} \beta_{v_i} + w(GR) \\ &\leq \sum_{i=1}^p \sum_{j=1}^{\lfloor c_i/2 \rfloor - 1} \sum_{v \in e_{i,j+1}} 2w(G_{i,j}) \\ &\quad + \sum_{i \in S_{\text{odd}}} 2w(G_{i,\lfloor c_i/2 \rfloor}) + w(GR) \\ &\leq \sum_{i=1}^p \sum_{j=1}^{\lfloor c_i/2 \rfloor - 1} 4w(G_{i,j}) + 3w(GR) \\ &\leq 4 \sum_{j=1}^{q-1} W_j + 3w(GR). \quad (3) \end{aligned}$$

The equality holds since the summation over all β_v adds up the total weight of edges in OPT except for those in GR . The first inequality follows from Lemma 4. The second inequality follows from the definition of GR and since each $e_{i,j}$ has exactly two vertices. The third inequality is proved as follows: Denote by $v_{i,j}$ the number of times, from the first j iterations, that cluster i was active.² Denote $G_j = \bigcup G_{i,v_{i,j}}$ (where $G_{i,0} = \emptyset$). Thus, G_j is an m_j -matching that contains from each cluster a greedy matching with the same number of vertices as it contains after the j th iteration of Algorithm *Metric*. Since G_j is an m_j -matching, and M_{m_j} is a maximum weight m_j -matching, $w(G_j) \leq w(M_{m_j}) = W_j$. Therefore, $\sum_{i=1}^p \sum_{j=1}^{\lfloor c_i/2 \rfloor - 1} w(G_{i,j}) = \sum_{j=1}^{q-1} w(G_j) \leq \sum_{j=1}^{q-1} W_j$.

Using Lemma 1, we get that

$$w(G_{i,\lfloor c_i/2 \rfloor}) \leq \frac{2}{c_i} w(O_i) \leq \frac{2}{k} w(O_i),$$

and therefore

$$w(GR) = \sum_{i=1}^p w(G_{i,\lfloor c_i/2 \rfloor}) \leq \frac{2}{k} opt.$$

Substitution in (3) gives $opt(1 - \frac{6}{k}) \leq 4 \sum_{j=1}^{q-1} W_j$. With (2) this gives $apx \geq (\frac{1}{2} - \frac{3}{k})opt$. \square

References

- [1] T. Feo, O. Goldschmidt, M. Khellaf, One half approximation algorithms for the k -partition problem, *Oper. Res.* 40 (1992) S170–S172.
- [2] T. Feo, M. Khellaf, A class of bounded approximation algorithms for graph partitioning, *Networks* 20 (1990) 181–195.
- [3] R. Hassin, S. Rubinstein, Robust matchings, *SIAM J. Discrete Math.* 15 (2002) 530–537.
- [4] R. Hassin, S. Rubinstein, Approximation algorithms for the metric maximum clustering problem with given cluster sizes, *Oper. Res. Lett.* 31 (2003) 179–184.
- [5] R. Hassin, S. Rubinstein, An approximation algorithm for maximum triangle packing, in: Proc. ESA 2004, in: *Lecture Notes in Comput. Sci.*, vol. 3221, Springer, Berlin, 2004, pp. 395–402.
- [6] R. Hassin, S. Rubinstein, A. Tamir, Approximation algorithms for maximum dispersion, *Oper. Res. Lett.* 21 (1997) 133–137.

² For example, in Fig. 1 we have $v_{1,1} = 1$ but $v_{2,1} = v_{3,1} = v_{4,1} = 0$. Also, $v_{1,2} = 2, v_{2,2} = 1$, etc.