Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Operations Research Letters 31 (2003) 179–184
Operations Research Letters
www.elsevier.com/locate/dsw

Approximation algorithms 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 4 November 2002; accepted 3 December 2002

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 $c_1,\dots,c_p$ that sum to $|V|$. The goal is to find a partition of V into disjoint clusters of sizes $c_1,\dots,c_p$, that maximizes the sum of weights of edges whose two ends belong to the same cluster. We describe approximation algorithms for this problem.

Keywords: Approximation algorithms; Maximum 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 non-negative edge weights $w(i,j)$, $(i,j)\in E$, that satisfy the triangle inequality. In the *general case* of the problem, cluster sizes $c_1 \geqslant c_2 \geqslant \cdots \geqslant c_p \geqslant 1$ such that $c_1+\cdots+c_p=n$ are given. In the *uniform case*, $c_1=c_2=\cdots=c_p$. The problem is to partition V into sets of the given sizes, so that the total weight of edges inside the clusters is maximized. See [6] and its references for some applications.

Hassin and Rubinstein [3] gave a approximation algorithm whose error ratio is bounded by $1/2\sqrt{2}\approx 0.353$ for the general problem. We improve this result for the case in which cluster sizes are large. In particular, when the minimum cluster size increases, the performance guarantee of our algorithm increases asymptotically to 0.375.

Feo and Khellaf [2] treated the uniform case and developed a polynomial algorithm whose error ratio is bounded by $c/2(c-1)$ or $(c+1)/2c$, where the cluster size is $c=n/p$, and it is even or odd, respectively. The bound decreases to 1/2 as c approaches ∞. The algorithm's time complexity is dominated by computation of a maximum weight perfect matching. (Without the triangle inequality assumption, the bound is $1/(c-1)$ or $1/c$, respectively, but Feo, Goldschmidt and Khellaf [1] improved the bound to $\frac{1}{2}$ in the cases of $c=3$ and 4.) We describe an alternative algorithm for the uniform case that achieves the ratio of 1/2 and has a lower $O(n^2)$ complexity.

Hassin, Rubinstein and Tamir [4] generalized the algorithm of [2] and obtained a bound of $\frac{1}{2}$ for computing k clusters of size c each $(1\leqslant k\leqslant n/c)$ with maximum total weight. Our discussion

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

0167-6377/03/$ - see front matter
doi:10.1016/S0167-6377(02)00235-3

concerning the uniform case does not apply to this generalization.

For $E' \subset E$ we denote by $w(E')$ the total weight of edges in E'. For $V' \subseteq V$ we denote by $E(V')$ the edge set of the subgraph induced by V'. To simplify the presentation, we denote the weight $w(E(V'))$ of the edges in the subgraph induced by a vertex set V' by $w(V')$. We denote by *opt* the optimal solution value, and by *apx* the approximate value returned by a given approximation algorithm. A *p-matching* is a set of p vertex-disjoint edges in a graph. A p-matching with $p = \lfloor n/2 \rfloor$ is called *perfect*. 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. A *greedy perfect matching* has $p = \lfloor n/2 \rfloor$.

2. A $\frac{3}{8}$-approximation algorithm

Lemma 1. *Let M^g be a greedy k-matching. Let M' be an arbitrary 2k-matching. Then, for $i = 1, \ldots, k$, the weight of the ith largest edge in M^g is greater than or equal to the weight of the $(2i-1)$st largest edge in M'.*

Proof. Let $e_1, \ldots, e_k$ be the edges of M' in non-increasing order of weight. By the greedy construction, every edge of $e' \in M' \setminus M^g$ is incident to an edge of $e \in M^g$ with $w(e) \geqslant w(e')$. Since every edge of M^g can take the above role at most twice, it follows that for $e_1, \ldots, e_{2i-1}$ we use at least i edges of M^g all of which are at least as large as $w(e_{2i-1})$. □

Lemma 2. *If a cluster $C \subset V$ of size $|C| = c$ contains a k-matching M of weight W, then $w(C) \geqslant (c-k)W$.*

Proof. Let $V(M)$ be the set of vertices of the edges in M. Then, $|V(M)| = 2k$. Consider an edge $e = (u, v) \in M$. By the triangle inequality, for every vertex x, $w(u,x) + w(v,x) \geqslant w(u,v)$. Summing over $x \in C \setminus V(M)$, the total weight of the edges of C that connect u and v with vertices in $C \setminus V(M)$ is at least $(c-2k)w(u,v)$. Summation over M gives a weight of at least $(c-2k)W$.

A similar summation over $x \in V(M)$ (including $x = u, v$) gives a total weight of $2kw(u,v)$. However, every edge that contributes to this sum (including (u,v)) is counted twice. Hence, summation over M only gives that the total weight of these edges is at least kW.

Altogether, $w(C) \geqslant [(c-2k)+k]W = (c-k)W$. □

We first present our algorithm assuming that the cluster sizes are divisible by 4. After analyzing this case, we will show how to modify the algorithm for the general case, and obtain the same bound asymptotically for big cluster sizes.

Theorem 1. *Let apx be the total weight of edges in the clusters returned by Algorithm metric* (Fig. 1). *Then,*

$$apx \geqslant \frac{3}{8} opt.$$

Proof. Consider an optimal partition $O_1, \ldots, O_p$. Let M_i be a maximum matching in the subgraph induced by O_i, $i = 1, \ldots, p$. Let $M = M_1 \cup \cdots \cup M_p$ (note that $|M_i| = \frac{1}{2}c_i$ and $|M| = \frac{1}{2}|V|$). Let M'_1 be the matching consisting of the $\frac{1}{2}c_1$ heaviest edges in M, M'_2 the next $\frac{1}{2}c_2$ heaviest edges, and so on up to M'_p which consists of the $\frac{1}{2}c_p$ lightest edges in M. From the assumption that $c_1 \geqslant \cdots \geqslant c_p$ it follows that $\sum_i c_i w(M_i) \leqslant \sum_i c_i w(M'_i)$. The edge set of $E(O_i)$ can be covered by a set of $c_i - 1$ disjoint matchings. Since M_i is a maximum matching in O_i it follows that $w(O_i) \leqslant (c_i - 1)w(M_i)$ and therefore

$$opt = \sum_{i=1}^{p} w(O_i) \leqslant \sum_{i=1}^{p} c_i w(M_i) \leqslant \sum_{i=1}^{p} c_i w(M'_i).$$

Let M_i^g be the greedy matching's edges whose end vertices were inserted by Algorithm metric to S_i. Note that $|M_i^g| = c_i/4$.

By Lemma 1, $w(M'_i) \leqslant 2w(M_i^g)$. Therefore, by Lemma 2,

$$\begin{aligned} apx &\geqslant \sum_{i=1}^{p} (c_i - |M_i^g|) w(M_i^g) \\ &= \frac{3}{4} \sum_{i=1}^{p} c_i w(M_i^g) \\ &\geqslant \frac{3}{8} \sum_{i=1}^{p} c_i w(M'_i) \geqslant \frac{3}{8} opt. \end{aligned}$$ □

Metric
input
1. A complete undirected graph $G = (V, E)$ with weights $w(e)$, $e \in E$ satisfying the triangle inequality.
2. Constants $c_1 \geq \cdots \geq c_p \geq 4$ such that $c_i \equiv 0 \bmod 4$, and $\sum_i c_i \leq |V|$.
returns
Clusters $S_1, \ldots, S_p$ such that $|S_i| = c_i$.
begin
$m := \frac{|V|}{4}$.
$M^g = (e_1, \ldots, e_m) :=$ *a greedy m-matching.*
for every $i = 1, \ldots, p$
$S_i :=$ *the end vertices of the* $\frac{c_i}{4}$ *heaviest edges in* M^g.
$M^g := M^g \setminus S_i$.
Complete every S_i, $i = 1, \ldots, p$, to size c_i by adding arbitrary, yet unassigned vertices.
return $S_1, \ldots, S_p$.
end *Metric*

Fig. 1. Algorithm metric.

The situation is more complex when the restriction that the cluster sizes are divisible by 4 is removed. We propose to apply Algorithm *Metric* with sizes $c'_1, \ldots, c'_p$, where $c'_i = 4\lfloor c_i/4 \rfloor$, and complete the clusters arbitrarily to sizes $c_1, \ldots, c_p$. If the cluster sizes are large enough, the bound is not greatly affected by this change and will be asymptotically $\frac{3}{8}$. We now describe the changes in the algorithm and analysis that are necessary to account for any cluster sizes at least 4.

Suppose that for $i = 1, \ldots, p$, $c_i = 4k_i + \delta_i$, where $k_i \geqslant 1$ is an integer and $\delta_i \in \{0, 1, 2, 3\}$. Let O_i be the ith cluster in an optimal solution and let O'_i be a maximum weight subset of O_i resulting from the deletion of δ_i vertices. Then

$$\frac{w(O'_i)}{w(O_i)} \geqslant \frac{|E(O'_i)|}{|E(O_i)|} = \frac{\binom{4k_i}{2}}{\binom{c_i}{2}}$$

and therefore

$$opt = \sum_{i=1}^{p} w(O_i) \leqslant \sum_{i=1}^{p} \frac{\binom{c_i}{2}}{\binom{4k_i}{2}} w(O'_i). \qquad (1)$$

Since $|O'_i| = 4k_i$, $E(O'_i)$ can be covered by $4k_i - 1$ disjoint matchings. Hence, a maximum matching, M_i, in the subgraph induced by O'_i satisfies $w(O'_i) \leqslant (4k_i - 1)w(M_i)$ and with (1)

$$opt \leqslant \sum_{i=1}^{p} f_i w(M_i), \qquad (2)$$

where $f_i \equiv f(c_i) = (4k_i - 1)\binom{c_i}{2} \Big/ \binom{4k_i}{2} = c_i(c_i - 1)/4k_i$. Note that f is not necessarily monotone increasing. Let $j_1, \ldots, j_p$ be a permutation of $1, \ldots, p$ such that $f_{j_1} \geqslant \cdots \geqslant f_{j_p}$.

Let $M = M_1 \cup \cdots \cup M_p$. Note that $|M_i| = 2k_i$. Let M'_1 be the matching consisting of the $2k_{j_1}$ heaviest edges in M, M'_2 the next $2k_{j_2}$ heaviest edges, and so on up to M'_p which consists of the $2k_{j_p}$ lightest edges in M. Clearly,

$$\sum_{i=1}^{p} f_i w(M_i) \leqslant \sum_{i=1}^{p} f_{j_i} w(M'_{j_i}). \qquad (3)$$

Consider a greedy matching M^g of size $\frac{1}{2}|M|$. Let $M^g_{j_1}$ be the matching consisting of the k_{j_1} heaviest edges in M^g, $M^g_{j_2}$ the next k_{j_2} heaviest edges, and so on. Let S^g_i be the set of end vertices of the edges in M^g_i, and let $\{S_1, \ldots, S_p\}$ be an arbitrary completion of $(S^g_1, \ldots, S^g_p)$ to disjoint clusters of sizes $(c_1, \ldots, c_p)$.

By Lemma 1

$$w(M'_{j_i}) \leqslant 2w(M_i^g), \quad (4)$$

and by Lemma 2

$$w(S_i) \geqslant (c_i - k_i)w(M_i^g). \quad (5)$$

Combining (2)–(5) we obtain

$$opt \leqslant \sum_{i=1}^{p} g_i w(S_i),$$

where $g_i \equiv g(c_i) = 2f_i/(c_i - k_i) = c_i(c_i - 1)/2k_i(c_i - k_i)$.

Let $\beta = [\max\{g(c_i) \mid i = 1,\dots,p\}]^{-1}$, then

$$apx = \sum_{i=1}^{p} w(S_i) \geqslant \beta opt.$$

For example, for $c_i = 4,\dots,12$ $1/g_i$ is 0.5, 0.4, 0.333, 0.286, 0.428, 0.389, 0.356, 0.327, and 0.409. The bound improves over the previously known bound of $\frac{1}{2\sqrt{2}} \approx 0.353$ if $c_i \notin \{2,3,6,7,11,15,19\}$, and in particular if all cluster sizes are at least 20. When $\min\{c_i: i = 1,\dots,p\} \to \infty$, $\beta \to \frac{3}{8}$.

3. The uniform case

We now consider the uniform case, that is $c_i = c$ for $i = 1,\dots,p$. Consider the set of partitions of V into clusters of size c each. A *random solution* is obtained by randomly (uniformly) selecting such a partition. The following theorem states a bound on the expected value of a random solution. When the cluster sizes are not identical, Example 1 in Section 4 shows that the expected weight of a random solution is not a good approximation. Also note that in contrast to a similar bound for the related MAXIMUM CUT PROBLEM, our result requires that the edge weights satisfies the triangle inequality.

Theorem 2. *The expected weight of a random solution is at least* $\frac{1}{2}opt$.

Proof. Consider an optimal partition $OPT = (O_1,\dots,O_p)$. Let M be a matching consisting of the union of maximum perfect matchings in the subgraphs induced by $O_1,\dots,O_p$.

Suppose first that c is even. Then M is a perfect matching in G. Let S_v be the (random) set of edges that are incident to v in the cluster that contains v in a random solution. The weight of a random solution is

$$\frac{1}{2}\sum_{v\in V} w(S_v) = \frac{1}{2}\sum_{(u,v)\in M} [w(S_u) + w(S_v)].$$

Consider an edge $(u,v) \in M$. A solution in which u and v are contained in the same cluster satisfies, by the triangle inequality, that the average weight of an edge in $S_u \cup S_v$ is at least $\frac{1}{2}w(u,v)$. As for the solutions in which u and v belong to distinct clusters, we pair these solutions so that each pair consists of a solution and another one obtained by swapping u and v. Again it follows from the triangle inequality that the average weight of an edge in $S_u \cup S_v$ in such a pair of solutions is at least $\frac{1}{2}w(u,v)$. Hence the expected average weight of an edge in a random solution is at least 1/2 the average edge weight in M. The claim follows by observing that the average edge weight in M is at least as large as the average edge weight in OPT (see for example [4]).

Suppose now that c is odd. Then, M leaves out one vertex from each subset. We consider the stars incident to these vertices in the optimal and random solutions. Since M is the union of maximum matchings, the sum of these stars in OPT is at most $2w(M)$. On the other hand, from the same pairing argument and the triangle inequality, the expected total weight of these stars in a random solution is at least $w(M)$. The rest of the proof of this case follows the same arguments as when c was assumed to be even. □

We use the method of conditional probabilities [5] to de-randomize the algorithm while preserving its performance guarantee. At a given stage we have already determined the contents of several clusters and we deal with one *active* cluster that may now be partially filled. Let r be the number of clusters yet to be constructed excluding the active one. Let A be the vertices already assigned to the active cluster, $a = |A|$, and let B be the yet undecided vertices, $b = |B|$. We maintain and update for each $u \in B$ $\alpha_u = \sum_{v\in A} w(u,v)$, $\alpha = \sum_{u\in B} \alpha_u$ and $\beta = \sum_{u,v\in B} w(u,v)$. Altogether, these updates take $O(n^2)$ time.

The expected weight of a random completion of the assignment of vertices to clusters is

$$\frac{c-a}{b}\alpha + \frac{\binom{c-a}{2} + r\binom{c}{2}}{\binom{b}{2}}\beta.$$

The first term is the expected weight of edges between vertices that will be added to the active cluster and the vertices already in it, the second is the expected weight of edges between vertices in B that will be placed in the same cluster.

At each step we examine the insertion of each $u \in B$ to the active cluster (contributing α_u) followed by a random completion, and select the one that gives maximum expected weight. There are n insertions and each requires $O(n)$ examinations. Thus, altogether the complexity is $O(n^2)$.

4. Some bad examples

In this section, we propose some natural algorithms and provide for each of them an instance for which the algorithm performs badly.

In Section 3 we have shown that the expected weight of a random solution is at least $\frac{1}{2}opt$ when the clusters have a common size. The following example shows that when the cluster sizes are not identical the expected weight of a random solution may be very small relative to opt.

Example 1. Consider an instance with weights $w(1,j) = 1$, $j = 2,\ldots,n$, and $w(i,j) = 0$ otherwise. Let $c_1 = 2$ and $c_i = 1$, $i = 2,\ldots,n-1$. Then, only solutions in which vertex 1 is placed in the large cluster have a positive weight. This happens with probability $\frac{2}{n}$. Thus, $opt = 1$ whereas the expected weight of a random solution is less than $2/n$, and the ratio can be made arbitrarily small.

Next, we show that a natural local search approach does not guarantee a constant bound, even in the uniform case.

Example 2. We construct an instance with 0/1 weights. The vertex set V is composed of disjoint subsets $V_0, V_1, \ldots, V_\ell$. $|V_0| = (\ell - k)\ell$, and for $i = 1,\ldots,\ell$, V_i has a distinguished vertex $v_i \in V_i$, and $|V_i| = k$. Thus, $|V| = \ell^2$. The subgraph induced by the unit weight edges consists of the cliques induced by V_i, $i = 1,\ldots,\ell$, and the clique induced by the distinguished vertices $v_1,\ldots,v_\ell$. (Vertices in V_0 are not adjacent to any edge with unit weight.) Suppose that V must be partitioned into ℓ clusters of size ℓ each, and that $1 \ll k^2 \ll \ell \ll |V|$. The optimal solution contains a cluster with the distinguished vertices and its weight is of order ℓ^2. A solution in which each V_i, $i = 1,\ldots,\ell$ is contained in a different cluster cannot be improved by relocating less than k vertices, and its weight is $O(\ell k^2)$.

A perfect matching M that, for each $p = 1,\ldots,|M|$, contains p edges whose total weight is at least α times the maximum weight of a p-matching is said to be *α-robust*. Hassin and Rubinstein [3] proved that there always exists a $1/\sqrt{2}$-robust matching, and that there are instances where a higher robustness is not possible.

Consider the following extension of the algorithm of Feo and Khellaf [2] and Hassin et al. [4]:

1. Compute an α-robust matching S.
   Let $S = \{(u_j, v_j)\ \ j = 1,\ldots,m\}$, where $w(u_j,v_j) \geqslant w(u_{j+1},v_{j+1})\ j = 1,\ldots,m-1$.
2. For $j = 1,\ldots,p$, let $d_j = \lfloor c_j/2 \rfloor$ and $D_j = d_1 + \cdots + d_j\ j = 1,\ldots,p$. Let $D_0 = 0$. Set $V_i = \{u_j, v_j \mid j = D_{i-1}+1,\ldots,D_i\}\ i = 1,\ldots,p$.
3. For each i such that c_i is odd, add to V_i an arbitrary yet unassigned vertex.

Hassin and Rubinstein [3] proved that the algorithm returns an $\alpha/2$-approximation for the maximum clustering problem with cluster sizes $c_1,\ldots,c_p$. Thus, the best bound derived this way is $1/2\sqrt{2}$. Moreover, since both a maximum matching and a greedy perfect matching are $\frac{1}{2}$-robust, using such matchings in the algorithm guarantees at least a $\frac{1}{4}$-approximation.

The following example demonstrates that the $\frac{1}{4}$ bound is the best possible when using a maximum matching:

Example 3. Let $V = A \cup B \cup C \cup D \cup F$ where $|A| = M$, $|B| = |C| = |D| = |F| = M/2$, $D = \{d_1,\ldots,d_{M/2}\}$, and

$F = \{e_1, \ldots, e_{M/2}\}$. Let

$$w(i,j) = \begin{cases} 2, & i \in B \; j \in C, \\ 1, & i \in B \; j \notin C \text{ or } i \in C \; j \notin B, \\ 1, & (i,j) = (d_l, e_l) \; l = 1, \ldots, e_{M/2}, \\ 0, & i,j \in A, \\ \frac{1}{2} & \text{otherwise.} \end{cases}$$

Let $c = (M, 1, \ldots, 1)$ where the number of clusters of size 1 is $2M$, so that $\sum_i c_i = 3M = |V|$. An optimal solution will choose for the big cluster the set $B \cup C$ and thus

$$opt = 2(M)^2.$$

The edges $\{(d_1, e_1), \ldots, (d_{M/2}, e_{M/2})\}$ completed by arbitrary disjoint edges between A and $B \cup C$ constitute a maximum matching. The algorithm may choose such a matching and then have $D \cup F$ as the large cluster, and then

$$apx = \frac{1}{2}\left(\frac{M}{2}\right)^2 + \frac{M}{4}.$$

This gives asymptotically a bound of $\frac{1}{4}$.

We observe that a greedy matching may have an advantage over a maximum matching since it chooses larger edges for the large clusters. The following example shows however that the choice of a greedy matching does not guarantee more than a $\frac{3}{8}$-approximation, even for the case of two uniform clusters.

Example 4. Let $V = A \cup B \cup C$ with $|A| = |B| = M/2$, $|C| = M$, $A = \{a_1, \ldots, a_{M/2}\}$, and $B = \{b_1, \ldots, b_{M/2}\}$. Let $c = (M, M)$, and

$$w(i,j) = \begin{cases} 2, & i \in C \; j \in A \cup B, \\ 2, & (i,j) = (a_l, b_l) \; l = 1, \ldots, M/2, \\ \frac{4}{3}, & i,j \in A \text{ or } i,j \in B, \\ \frac{2}{3}, & i \in A \; j \in B \text{ or } i \in B \; j \in A, \\ 0, & i,j \in C. \end{cases}$$

The algorithm may choose the edges (a_i, b_i) for the greedy matching, resulting in clusters $A \cup B$ and C. The weight of this solution is

$$apx = \frac{2}{3}\frac{M^2}{2} + 2 \times \frac{4}{3}\binom{\frac{M}{2}}{2} + \frac{M}{2} \times 2 \approx \frac{M^2}{2}.$$

Consider a solution with sets $A \cup C_1$ and $B \cup C_2$, where $C_1, C_2 \subset C$ are disjoint and of cardinality $M/2$ each. The value of this solution is $2\left[2\left(\frac{M}{2}\right)^2 + \frac{4}{3}\binom{\frac{M}{2}}{2}\right] \approx \frac{4}{3}M^2$. Therefore the algorithm achieves in this case asymptotically no more than $\frac{3}{8}opt$.

References

[1] T. Feo, O. Goldschmidt, M. Khellaf, One half approximation algorithms for the κ-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, A. Tamir, Approximation algorithms for maximum dispersion, Oper. Res. Lett. 21 (1997) 133–137.

[5] P. Raghavan, Probabilistic construction of deterministic algorithms: approximating packing integer programs, J. Comput. System Sci. 37 (1988) 130–143.

[6] R.R. Weitz, S. Lakshminarayanan, An empirical comparison of heuristic methods for creating maximally diverse groups, J. Oper. Res. Soc. 49 (1998) 635–646.