数理解析研究所講究録
第 1554 巻 2007 年 71-77

Approximation Algorithm for Maximum Triangle Packing and Metric Maximum Clustering

Graduate School of Science and Engineering, Tokyo Denki University
Zhi-Zhong Chen Ruka Tanahashi Tatsuyuki Sekiyama

Abstract

This paper deals with the metric maximum clustering problem with given cluster sizes and the maximum triangle packing problem. For the former problem, Hassin and Rubinstein gave a randomized polynomial-time approximation algorithm achieving an expected ratio of $\frac{1}{2}-\frac{3}{k}$, where k is the size of the smallest cluster. We improve the ratio to $\frac{1}{2}-\frac{2}{k}+\frac{1}{k(k-1)}$ and also derandomize it. For the latter problem, Hassin and Rubinstein gave a randomized polynomial-time approximation algorithm achieving an expected ratio of $\frac{43}{83}(1-\epsilon)$. we improve the expected ratio of $\frac{187+320p}{347+640p}\cdot(1-\epsilon)$ for any constant $\epsilon>0$. Note that p is close to 0.27.

1 Introduction

In the *metric maximum clustering problem with given cluster sizes* (Metric MCP-GCS for short), we are given an edge-weighted complete graph $G=(V,E)$ and a sequence of positive integers $c_1,\ldots,c_p$ such that the edge weights are nonnegative and satisfy the triangle inequality and $\sum_{i=1}^{p} c_i=|V|$. The objective is to find a partition of V into disjoint clusters of sizes $c_1,\ldots,c_p$ such that the total weight of edges whose endpoints belong to the same cluster is maximized. This problem has a lot of applications [9] and a number of approximation algorithms known have been given for it and its special cases [1,2,7,3,4,5]. In particular, Hassin and Rubinstein [5] gave a randomized polynomial-time approximation algorithm for Metric MCP-GCS which achieves an expected ratio of $\frac{1}{2}-\frac{3}{k}$, where k is the size of the smallest cluster. In this paper, we modify and derandomize their algorithm to obtain a polynomial-time approximation algorithm for Metric MCP-GCS which achieves a ratio of $\frac{1}{2}-\frac{2}{k}+\frac{1}{k(k-1)}$. To our knowledge, our algorithm achieves the best ratio when k is large.

A problem closely related to Metric MCP-GCS is the *maximum triangle packing problem* (MTP for short). In this problem, we are given an edge-weighted complete graph $G=(V,E)$ such that the edge weights are nonnegative and $|V|$ is a multiple of 3. The objective is to find a partition of V into $|V|/3$ disjoint subsets each of size exactly 3 such that the total weight of edges whose endpoints belong to the same cluster is maximized. Obviously, if we do not require that the edge weights satisfy the triangle inequality in Metric MCP-GCS, then MTP becomes a special case of Metric MCP-GCS. Hassin and Rubinstein [4] gave a randomized polynomial-time approximation algorthm for MTP and claimed that their algorithm achieves an expected ratio of $\frac{89}{169}(1-\epsilon)$ for any constant $\epsilon>0$. However, the third author of this paper pointed out a flaw in their analysis and they [6] have corrected the ratio to $\frac{43}{83}(1-\epsilon)$. In this paper, we modify their algorithm to obtain a polynomial-time approximation algorithm for MTP which achieves an expected ratio of $\frac{187+320p}{347+640p}\cdot(1-\epsilon)>\frac{88.85}{169}\cdot(1-\epsilon)$. Note that p is close to 0.27.

2 Basic Definitions

Throughout the remainder of this paper, a graph means an undirected graph without parallel edges or self-loops whose edges each have a nonnegative weight.

Let G be a graph. We denote the vertex set of G by $V(G)$ and denote the edge set of G by $E(G)$. The *weight* of G, denoted by $w(G)$, is the total weight of edges in G. We denote the weight of an edge $e \in E(G)$ by $w_G(e)$. For a subset F of $E(G)$, we use $w_G(F)$ to denote the total weight of edges in F, and use $V(F)$ to denote the set of endpoints of the edges in F. The *weight* of a subgraph H of G, denoted by $w_G(H)$, is $w_G(E(H))$. The *degree* of a vertex v in G, denoted by $d_G(v)$, is the number of edges incident to v in G.

For a function b mapping each vertex v of G to a nonnegative integer, a *b-matching* of G is a subset F of $E(G)$ such that each vertex v of G is incident to at most $b(v)$ edges in F; moreover, a *maximum-weight b-matching* of G is a b-matching M of G such that $w_G(M) \geq w_G(M')$ for all b-matching M' of G. When $b(v) \leq 1$ for all vertices v of G, a b-matching of G is called a *matching* of G. For a natural number k, a matching M of G is called a *k-matching* of G if $|M| = k$, is called *a maximum-weight k-matching* of G if $w_G(M) \geq w_G(M')$ for all k-matchings M' of G, and is called *a perfect matching* of G if $2|V(M)| \geq |V(G)| - 1$.

A *cycle* in G is a connected subgraph of G in which each vertex is of degree 2. A *path* in G is either a single vertex of G or a connected subgraph of G in which exactly two vertices are of degree 1 and the others are of degree 2. The *length* of a cycle or path C, denoted by $|C|$, is the number of edges on C. A *Hamiltonian cycle* is a cycle C with $V(C) = V(G)$. A *cycle cover* of G is a subgraph H of G with $V(H) = V(G)$ in which each vertex is of degree 2.

For a sequence $c_1, \ldots, c_p$ of positive integers with $\sum_{i=1}^{p} c_i = |V(G)|$, a $(c_1, \ldots, c_p)$-*clustering* of G is a partition of $V(G)$ into disjoint subsets (called *clusters*) of sizes $c_1, \ldots, c_p$, respectively. The *weight* of a $(c_1, \ldots, c_p)$-clustering $\{C_1, \ldots, C_p\}$ of G is the total weight of edges $\{u, v\}$ of G such that some cluster C_i ($1 \leq i \leq p$) contains both u and v. A *triangle packing* of G is a $(3, \ldots, 3)$-clustering of G. Note that G has a triangle packing if and only if $|V(G)|$ is a multiple of 3.

For a random event A, $\Pr[A]$ denotes the probability that A occurs. For two random events A and B, $\Pr[A \mid B]$ denotes the probability that A occurs given the occurrence of B. For a random variable X, $\mathcal{E}[X]$ denotes the expected value of X.

3 Approximation Algorithm for Metric MCP-GCS

Throughout this section, fix an instance $(G, c_1, \ldots, c_p)$ of Metric MCP-GCS. Without loss of generality, we may assume that $c_1 \geq c_2 \geq \ldots \geq c_p$. Let $q = \lfloor \frac{c_1}{2} \rfloor$ and $S_{odd} = \{i \in \{1, \ldots, p\} \mid c_i \text{ is odd}\}$. We want to find a $(c_1, \cdots, c_p)$-clustering $\{C_1, \ldots, C_p\}$ of large weight. The following algorithm is for this purpose and is a derandomization of Hassin and Rubinstein's algorithm [5]:

(1) Initialize $C_1 = \cdots = C_p = \emptyset$, $a_1 = 2\lfloor \frac{c_1}{2} \rfloor, \ldots, a_p = 2\lfloor \frac{c_p}{2} \rfloor$, $m_0 = 0$, $M_0 = \emptyset$.
(*Comment:* $S_{odd} = \{i \in \{1, \ldots, p\} \mid a_i = c_i - 1\}$.)

(2) For $j = 1, \ldots, q$ (in this order), perform the following steps:

   (a) Let r_j be the maximum $i \in \{1, \ldots, p\}$ with $a_i = a_1$.

   (b) Let $m_j = m_{j-1} + r_j$.

   (c) Compute a maximum m_j-matching M_j of G with $V(M_{j-1}) \subseteq V(M_j)$.
   (*Comment:* By Lemma 2 in [5], this step can be done in polynomial time.)

(d) For each $i \in \{1, \ldots, r_j\}$, decrease a_i by 2.

(3) Arbitrarily distribute the edges in M_1 to $C_1, \ldots, C_{r_1}$ so that each C_i $(1 \le i \le r_1)$ receives (the endpoints of) exactly one edge in M_1.

(4) For $j = 2, \ldots, q$ (in this order), perform the following steps:

(a) Let $U_j = V(M_j) - V(M_{j-1})$.

(b) Construct a complete bipartite graph B_j as follows: The vertex set of B_j is $U_j \cup \{C_1, \ldots, C_{r_{j-1}}\}$. More precisely, the vertices on one side of B_j are exactly the vertices in U_j and the vertices on the other side of B_j are exactly the clusters $C_1, \ldots, C_{r_{j-1}}$. The weight of each edge (u, C_i) of B_j with $u \in U_j$ and $i \in \{1, \ldots, r_{j-1}\}$ is $\sum_{v \in C_i} w_G(\{u, v\})$.

(c) Compute a maximum-weight b-matching N_j in B_j, where $b(u) = 1$ for each $u \in U_j$ and $b(C_i) = 2$ for each $i \in \{1, \ldots, r_{j-1}\}$.
(*Comment:* Since B_j is complete and $r_j \ge r_{j-1}$, each C_i $(1 \le i \le r_{j-1})$ is incident to exactly two edges of N_j.)

(d) For each edge $(u, C_i) \in N_j$, add u to C_i.

(e) Arbitrarily distribute those vertices in U_j not incident to an edge in N_j to $C_{r_{j-1}+1}, \ldots, C_{r_j}$ so that each C_i $(r_{j-1} + 1 \le i \le r_j)$ receives exactly two vertices.

(5) Arbitrarily distribute the vertices in $V(G) - \bigcup_{1 \le i \le p} V(C_i)$ to the sets C_i with $i \in S_{odd}$ so that each such set C_i receives exactly one vertex.

(6) Output $C_1, \ldots, C_p$.

Lemma 3.1 *Let Apx be the weight of the clustering $C_1, \ldots, C_p$ output by the algorithm. Then, $Apx \ge 2 \sum_{j=1}^{q-1} w_G(M_j)$.*

Consider an optimal clustering $O_1, \ldots, O_p$ for $(G, c_1, \ldots, c_p)$. Let Opt be the weight of this clustering. For each $i \in \{1, \ldots, p\}$ such that c_i is odd, we choose a vertex $t_i \in O_i$ such that $\sum_{u \in O_i - \{t_i\}} w_G(\{t_i, u\}) \le \sum_{u \in O_i - \{v\}} w_G(\{v, u\})$ for all $v \in O_i$.

The following lemma is the key for us to improve the ratio obtained by Hassin and Rubinstein's algorithm:

Lemma 3.2 $\sum_{i=1}^{p} \sum_{u \in O_i - \{t_i\}} w_G(\{t_i, u\}) \le \frac{2}{k} Opt$, *where $k = \min\{c_1, \ldots, c_p\}$.*

For each $i \in \{1, \ldots, p\}$, let $O'_i = O_i$ if c_i is even, and let $O'_i = O_i - \{t_i\}$ otherwise. Let $Opt' = \sum_{i=1}^{p} \sum_{\{u,v\} \subseteq O'_i} w_G(\{u, v\})$.

Lemma 3.3 $Opt' \le 4 \sum_{j=1}^{q-1} w_G(M_j) + \frac{2}{k-1} Opt'$, *where k is as in Lemma 3.2.*

Theorem 3.4 *There is a polynomial-time approximation algorithm for* Metric MCP-GCS *that achieves a ratio of at least $\frac{1}{2} - \frac{2}{k} + \frac{1}{k(k-1)}$.*

4 An Approximation Algorithm

Throughout this section, fix an instance G of MTP and an arbitrary constant $\epsilon > 0$. Moreover, fix a maximum-weight triangle packing $\mathcal{O}pt$ of G.

To compute a triangle packing of large weight, Hassin and Rubinstein's algorithm [4] (H&R-algorithm for short) starts by computing a maximum-weight cycle cover $\mathcal{C}$ of G. It then breaks each cycle $C \in \mathcal{C}$ with $|C| > \frac{1}{\epsilon}$ into cycles of length at most $\frac{1}{\epsilon}$. This is done by removing a set F of edges on C with $w_G(F) \leq \epsilon w_G(C)$ and then adding one edge between each resulting path. In this way, the length of each cycle in $\mathcal{C}$ becomes short, namely, is at most $\frac{1}{\epsilon}$. H&R-algorithm then uses $\mathcal{C}$ to compute three triangle packings $P_1, ..., P_3$ of G, and further outputs the packing whose weight is maximum among the three.

P_1 is computed from $\mathcal{C}$ by a deterministic subroutine. Its weight is large when the total weight of edges in those cycles $C \in \mathcal{C}$ with $|C| = 3$ is large compared to the weight of $\mathcal{C}$. Here, instead of detailing how to compute P_1, we just mention the following result:

Lemma 4.1 [4] *Let $\alpha \cdot w_G(\mathcal{C})$ be the total weight of edges in those cycles $C \in \mathcal{C}$ with $|C| = 3$. Then, $w_G(P_1) \geq \frac{1+\alpha}{2} \cdot w_G(\mathcal{C}) \geq \frac{1+\alpha}{2}(1-\epsilon) \cdot w_G(\mathcal{O}pt)$.*

P_2 is also computed from $\mathcal{C}$ by a deterministic subroutine. Its weight is large when the total weight of those edges $\{u, v\}$ such that some cluster in $\mathcal{O}pt$ contains both u and v and some cycle in $\mathcal{C}$ contains both u and v is large compared to the weight of $\mathcal{C}$. Here, instead of detailing how to compute P_2, we just mention the following result:

Lemma 4.2 [4] *Let $\beta \cdot w_G(\mathcal{O}pt)$ be the total weight of those edges $\{u, v\}$ such that some cluster in $\mathcal{O}pt$ contains both u and v and some cycle in $\mathcal{C}$ contains both u and v. Then, $w_G(P_2) \geq \beta \cdot w_G(\mathcal{O}pt)$.*

Unlike P_1 and P_2, P_3 is computed from $\mathcal{C}$ by a complicated randomized subroutine. In Section 4.1, we substantially modify their subroutine, obtaining a new randomized subroutine for computing P_3. In Section 4.2, we analyze the approximation ratio achieved by the new algorithm.

4.1 Computation of P_3

Throughout this subsection, let p be the smallest real number satisfying the inequality $\frac{27}{20}p^2 - \frac{9}{10}p^3 \geq \frac{27}{320}$; the reason why we select p in this way will become clear in Lemma 4.8. Note that p is close to 0.27 and hence $p < \frac{1}{2}$. Let $C_1, ..., C_r$ be the cycles in $\mathcal{C}$. Consider the following randomized subroutine which computes P_3 from $\mathcal{C}$ as follows:

(1) Compute a maximum-weight b-matching M_1 in a graph G_1, where

- $V(G_1) = V(G)$,
- $E(G_1)$ consists of those $\{u, v\} \in E(G)$ such that u and v belong to different cycles in $\mathcal{C}$, and
- $b(v) = 2$ for each $v \in V(G_1)$.

(2) In parallel, for each cycle C_i in $\mathcal{C}$, process C_i by performing the following steps:

(a) Initialize R_i to be the empty set.

(b) If $|C_i| = 3$, then for each edge e of C_i, add e to R_i with probability p.
(*Comment:* $\mathcal{E}[w_G(R_i)] = (1-p) \cdot w_G(C_i)$. Moreover, each vertex of C_i is incident to exactly one edge of R_i with probability $2p(1-p)$. Furthermore, each vertex of C_i is incident to exactly two edges of R_i with probability p^2. Thus, each vertex of C_i is incident to at least one edge of R_i with probability $2p - p^2$.)

(c) If $|C_i| \geq 4$, then perform the following steps:

i. Choose one edge e_1 from C_i uniformly at random.

ii. Starting at e_1 and going clockwise around C_i, label the other edges of C_i as $e_2, \ldots, e_c$ where c is the number of edges in C_i.

iii. Add the edges e_j with $j \equiv 1 \pmod 4$ and $j \leq c-3$ to R_i.
(*Comment:* R_i is a matching of C_i and $|R_i| = \lfloor \frac{|C_i|}{4} \rfloor$.)

iv. If $c \equiv 1 \pmod 4$, then add e_{c-1} to R_i with probability $\frac{1}{4}$.
(*Comment:* R_i remains to be a matching of C_i. Moreover, $\mathcal{E}[|R_i|] = \frac{|C_i|-1}{4} + 1 \cdot \frac{1}{4} = \frac{|C_i|}{4}$.)

v. If $c \equiv 2 \pmod 4$, then add e_{c-1} to R_i with probability $\frac{1}{2}$.
(*Comment:* R_i remains to be a matching of C_i. Moreover, $\mathcal{E}[|R_i|] = \frac{|C_i|-2}{4} + 1 \cdot \frac{1}{2} = \frac{|C_i|}{4}$.)

vi. If $c \equiv 3 \pmod 4$ and $c > 3$, then add e_{c-2} to R_i with probability $\frac{3}{4}$.
(*Comment:* R_i remains to be a matching of C_i. Moreover, $\mathcal{E}[|R_i|] = \frac{|C_i|-3}{4} + 1 \cdot \frac{3}{4} = \frac{|C_i|}{4}$.)

(3) Let $R = R_1 \cup \cdots \cup R_r$.
(*Comment:* If $|C_i| = 3$, then $\Pr[e \in R_i] = p$ for every edge e of C_i. If $|C_i| \geq 4$, then $\mathcal{E}[|R_i|] = \frac{|C_i|}{4}$ by the comments on Step 2(c)iv through 2(c)vi. Moreover, each edge of C_i with $|C_i| \geq 4$ is added to R_i with the same probability. Thus, if $|C_i| \geq 4$, then $\Pr[e \in R_i] = \frac{1}{4}$ for every edge e of C_i, and hence each vertex of C_i is incident to at least one edge of R with probability $\frac{1}{2}$.)

(4) Let M_2 be the set of all edges $\{u, v\} \in M_1$ such that both u and v are of degree 0 or 1 in graph $\mathcal{C} - R$. Let G_2 be the graph $(V(G), M_2)$.

(5) For each odd cycle C of G_2, select one edge uniformly at random and delete it from G_2.

(6) Partition the edge set of G_2 into two matchings N_1 and N_2.

(7) For each edge e of G_2 which alone forms a connected component of G_2, add e to the matching $N_i (i \in \{1, 2\})$ which does not contain e.

(8) Select M from N_1 and N_2 uniformly at random.
(*Comment:* M is a matching of the graph $(V(G), M_1)$.)

(9) Let $\mathcal{C}'$ be the graph obtained from graph $\mathcal{C} - R$ by adding the edges in M.
(*Comment:* Each connected component of $\mathcal{C}'$ is a path or cycle. Moreover, each cycle K in $\mathcal{C}'$ may be a triangle or not. If K is a triangle, then it must be a triangle in $\mathcal{C}$. On the other hand, if K is not a triangle, then it must contain at least two edges in M.)

(10) Classify the cycles C of $\mathcal{C}'$ into three types: *superb*, *good*, or *ordinary*. Here, C is *superb* if $|C| = 3$; C is *good* if $|C| = 6$, $|E(C) \cap M| = 2$, and there are triangles C_i and C_j in $\mathcal{C}$

such that $|E(C_i) \cap E(C)| = 2$ and $|E(C_j) \cap E(C)| = 2$; C is *ordinary* if it is neither good nor superb.

(11) For each ordinary cycle C in $\mathcal{C}'$, choose one edge in $E(C) \cap M$ uniformly at random and delete it from $\mathcal{C}'$.

(12) For each good cycle C in $\mathcal{C}'$, change C back to two triangles in $\mathcal{C}$ as follows: Delete the two edges of $M \cap E(C)$ from C and then close each of the two resulting paths (of length 2) by adding the edge between its endpoints.
(*Comment:* Because of the maximality of $\mathcal{C}$, this step does not decrease $w_G(\mathcal{C}')$.)

(13) If $\mathcal{C}'$ has at least one path component, then connect the path components of $\mathcal{C}'$ into a single cycle Y by adding some edges of G, and further break Y into paths each of length 2 by removing a set F of edges from Y with $w_G(F) \le \frac{1}{3} \cdot w_G(Y)$.

(14) Let P_3 be the $(3, ..., 3)$-clustering of G induced by the connected components of $\mathcal{C}'$. More precisely, the clusters in P_3 one-to-one correspond to the vertex sets of the connected components of $\mathcal{C}'$.

Lemma 4.3 *For each* $e \in M_1$, $\Pr[e \in M |\ e \in M_2] \ge \frac{9}{20}$.

Lemma 4.4 *For each edge* $e \in M$ *such that at least one endpoint of* e *does not appear on a triangle in* $\mathcal{C}$, e *survives the deletion in Step 11 with probability at least* $\frac{3}{4}$.

Lemma 4.5 *For each* $e \in M_1$ *such that neither endpoint of* e *appears on a triangle in* $\mathcal{C}$, e *is contained in* $\mathcal{C}'$ *immediately after Step 11 with probability at least* $\frac{27}{320}$.

Lemma 4.6 *For each* $e \in M_1$ *such that exactly one endpoint of* e *appear on a triangle in* $\mathcal{C}$, e *is contained in* $\mathcal{C}'$ *immediately after Step 11 with probability at least* $\frac{27}{320}$.

Lemma 4.7 *Suppose that* $e = \{u_1, v_1\}$ *is an edge in* M *such that both* u_1 *and* v_1 *appear on triangle in* $\mathcal{C}$ *and both* u_1 *and* v_1 *are incident to exactly one edge in* R. *Then, the probability that* e *is contained in* $\mathcal{C}'$ *immediately after Step 11 is at least* $\frac{3}{4}$.

Lemma 4.8 *For each* $e \in M_1$ *such that both endpoints of* e *appear on triangles in* $\mathcal{C}$, e *is contained in* $\mathcal{C}'$ *immediately after Step 11 with probability at least* $\frac{27}{320}$.

4.2 Analysis of the Approximation Ratio

By the comment on Step 3, the expected total weight of the edges of $\mathcal{C}$ remaining in $\mathcal{C}'$ immediately after Step 11 is at least $\left((1-p)\alpha + \frac{3}{4}(1-\alpha)\right)w_G(\mathcal{C}) = \left(\frac{3}{4} - (p - \frac{1}{4})\alpha\right)w_G(\mathcal{C}) \ge \left(\frac{3}{4} - (p - \frac{1}{4})\alpha\right)(1-\epsilon)w_G(\mathcal{O}pt)$. Moreover, by Lemmas 4.5 through 4.8, the expected total weight of edges of M_1 remaining in $\mathcal{C}'$ immediately after Step 11 is at least $\frac{27}{320}w_G(M_1)$. Furthermore, by the construction of M_1, $w_G(M_1)$ is larger than or equal to the total weight of those edges $\{u, v\}$ such that some cluster in $\mathcal{O}pt$ contains both u and v but no cycle in $\mathcal{C}$ contains both u and v. So, $w_G(M_1) \ge (1-\beta)w_G(\mathcal{O}pt)$. Now, since $w_G(P_3)$ is at least $\frac{2}{3}$ of the total weight of edges in $\mathcal{C}'$ immediately after Step 11, we have

$$\mathcal{E}[w_G(P_3)] \geq \frac{2}{3}\Big(\frac{3}{4} - (p - \frac{1}{4})\alpha\Big)(1-\epsilon)w_G(\mathcal{O}pt) + \frac{2}{3} \cdot \frac{27}{320}(1-\beta)w_G(\mathcal{O}pt) \tag{3.1}$$

$$= \Big(\frac{89}{160} - \frac{1}{2}\epsilon - \frac{2}{3}(p - \frac{1}{4})(1-\epsilon)\alpha - \frac{9}{160}\beta\Big)w_G(\mathcal{O}pt). \tag{3.2}$$

So, by Lemma 4.1 and 4.2, we have

$$\frac{3}{4}(p - \frac{1}{4})w_G(P_1) + \frac{9}{160}w_G(P_2) + w_G(P_3) \geq \frac{187 + 320p - (320p + 160)\epsilon}{480} \cdot w_G(\mathcal{O}pt).$$

Therefore, the weight of the best packing among P_1, P_2, and P_3 is at least

$$\frac{187 + 320p - (320p + 160)\epsilon}{640p + 347} \cdot w_G(\mathcal{O}pt) \geq \frac{187 + 320p}{347 + 640p} \cdot (1-\epsilon)w_G(\mathcal{O}pt).$$

In summary, we have proven the following theorem:

Theorem 4.9 *For any constant $\epsilon > 0$, there is a polynomial-time randomized approximation algorithm for MTP that achieves an expected ratio of $\frac{187+320p}{347+640p} \cdot (1-\epsilon) > \frac{88.85}{169} \cdot (1-\epsilon)$.*

References

[1] T. Feo, O. Goldschmidt, and M. Khellaf. One Half Approximation Algorithms for the k-Partition Problem. *Operations Research* **40** (1992) S170-S172.

[2] T. Feo and M. Khellaf. A Class of Bounded Approximation Algorithms for Graph Partitioning. *Networks* **20** (1990) 185-195.

[3] R. Hassin and S. Rubinstein. Robust Matchings. *SIAM Journal on Discrete Mathematics* **15** (2002) 530-537.

[4] R. Hassin and S. Rubinstein. An Approximation Algorithm for Maximum Triangle Packing. *Discrete Applied Mathematics* **154** (2006) 971-979.

[5] R. Hassin and S. Rubinstein. An Improved Approximation Algorithm for the Metric Maximum Clustering Problem with Given Cluster Sizes. *Infomation Processing Letters* **98** (2006) 92-95.

[6] R. Hassin and S. Rubinstein. Erratum to "An approximation algorithm for maximum triangle packing" : [Discrete Applied mathematics 154 (2006) 971-979]. *Discrete Applied Mathematics* **154** (2006) 2620-2620.

[7] R. Hassin, S. Rubinstein, and A. Tamir. Approximation Algorithms for Maximum Dispersion. *Operations Research Letters* **21** (1997) 133-137.

[8] P. Raghavan. Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. *Journal of Computer and System Siences* **38** (1994) 683-707.

[9] R.R. Weitz and S. Lakshminarayanan. An Empirical Comparison of Heuristic Methods for Creating Maximally Diverse Groups. *Journal of the Operational Research Society* **49** (1998) 635-646.