An Improved Randomized Approximation Algorithm for Maximum Triangle Packing

Zhi-Zhong Chen[1], Ruka Tanahashi[1], and Lusheng Wang[2]

[1] Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan
[2] Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

Abstract. This paper deals with the maximum triangle packing problem. For this problem, Hassin and Rubinstein gave a randomized polynomial-time approximation algorithm and claimed that it achieves an expected ratio of $\frac{89}{169}(1-\epsilon)$ for any constant $\epsilon > 0$. However, their analysis was flawed. We present a new randomized polynomial-time approximation algorithm for the problem which achieves an expected ratio very close to their claimed expected ratio.

1 Introduction

In the *maximum triangle packing problem* (MTP for short), we are given an edge-weighted complete graph $G = (V, E)$ such that the edge weights are non-negative and $|V|$ is a multiple of 3. The objective is to find a partition of V into $\frac{1}{3}|V|$ disjoint subsets each of size exactly 3 such that the total weight of edges whose endpoints belong to the same subset is maximized. MTP is a classic NP-hard problem; indeed, it is contained in Garey and Johnson's famous book on the theory of NP-completeness [2]. MTP is not only NP-hard but also MAX SNP-hard [5], implying that it does not admit a polynomial-time approximation scheme unless $P = NP$. A stronger hardness result has been obtained by Chlebík and Chlebíková [1]: No polynomial-time approximation algorithm can approximate MTP within a ratio of 0.9929 unless $P = NP$.

On the positive side, Hassin and Rubinstein [3] have presented a randomized polynomial-time approximation algorithm for MTP. In [3], they claimed that their algorithm achieves an expected ratio of $\frac{89}{169}(1-\epsilon)$ for any constant $\epsilon > 0$. However, the first author of this paper pointed out a flaw in their analysis to them and they [4] have corrected the expected ratio to $\frac{43}{83}(1-\epsilon)$.

In this paper, we obtain a new randomized polynomial-time approximation algorithm for MTP by substantially modifying the algorithm due to Hassin and Rubinstein. Like their algorithm, our algorithm starts by computing a maximum cycle cover $\mathcal{C}$ in the input graph G, then processes $\mathcal{C}$ to obtain three triangle packings of G, and finally outputs the maximum weighted packing among the three packings. Unlike their algorithm, our algorithm processes triangles in $\mathcal{C}$ in a different way than the other cycles in $\mathcal{C}$, and tries to connect the cycles in $\mathcal{C}$ by

R. Fleischer and J. Xu (Eds.): AAIM 2008, LNCS 5034, pp. 97–108, 2008.

using some edges in a maximum-weight b-matching (rather than a maximum-weight matching) between the cycles. Although our algorithm may look similar to the one in [3], our algorithm needs a deeper analysis of various probabilities. By carefully analyzing the new algorithm, we show that it achieves an expected ratio of $\frac{88.85}{169}(1-\epsilon)$ for any constant $\epsilon > 0$. Although the new ratio (namely, $\frac{88.85}{169}(1-\epsilon)$) may seem to be only slightly better than the old ratio (namely, $\frac{43}{83}(1-\epsilon)$), it is of interest for the following two reasons: First, the new ratio is very close to $\frac{89}{169}(1-\epsilon)$ which is the expected ratio wrongly claimed in [3]; hence our new algorithm and its analysis can be viewed as an almost complete correction of the flaw committed in [3]. Second, the improvement (achieved by our new algorithm) from $\frac{43}{83}(1-\epsilon)$ to $\frac{88.85}{169}(1-\epsilon)$ is almost half the improvement (achieved by the algorithm in [3]) from the trivial ratio $\frac{1}{2}$ to $\frac{43}{83}(1-\epsilon)$.

Our randomized algorithm is too sophisticated to derandomize. However, we can modify it to obtain another randomized algorithm which achieves a slightly smaller expected ratio but can be derandomized using the pessimistic estimator method [6]; the resulting deterministic polynomial-time approximation algorithm for MTP still achieves a better ratio (namely, $\frac{43.1}{83}(1-\epsilon)$) than the *expected* ratio achieved by Hassin and Rubinstein's *randomized* algorithm [4]. We omit the details here.

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. A graph G has a vertex set $V(G)$ and an edge set $E(G)$. We denote the weight of a subgraph H of G by $w(H)$. 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. A *path component* of G is a connected component of G that is a path.

For a random event A, $\Pr[A]$ denotes the probability that A occurs. For a random event A and one or more random events $B_1, \ldots, B_h$, $\Pr[A \mid B_1, \ldots, B_h]$ denotes the probability that A occurs given the occurrences of $B_1, \ldots, B_h$. For a random variable X, $\mathcal{E}[X]$ denotes the expected value of X. For a random variable X and one or more random events $B_1, \ldots, B_h$, $\mathcal{E}[X \mid B_1, \ldots, B_h]$ denotes the expected value of X given the occurrences of $B_1, \ldots, B_h$.

3 Sketch of Hassin and Rubinstein's 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 [3] (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 in C with

$w(F) \le \epsilon \cdot w(C)$ and then adding one edge between the endpoints of 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_2, and P_3 of G, and further outputs the packing whose weight is maximum among the three.

Computing P_1 and P_2 from $\mathcal{C}$ in H&R-algorithm is easy:

Lemma 1. [3] *Let $\alpha \cdot w(\mathcal{C})$ be the total weight of edges in triangles in $\mathcal{C}$. Then, $w(P_1) \ge \frac{1+\alpha}{2} \cdot w(\mathcal{C}) \ge \frac{1+\alpha}{2}(1-\epsilon) \cdot w(\mathcal{O}pt)$.*

Lemma 2. [3] *Let $\beta \cdot w(\mathcal{O}pt)$ be the total weight of those edges $\{u, v\}$ such that some triangle in $\mathcal{O}pt$ contains both u and v and some cycle in $\mathcal{C}$ contains both u and v. Then, $w(P_2) \ge \beta \cdot w(\mathcal{O}pt)$.*

Unlike P_1 and P_2, the computation of P_3 from $\mathcal{C}$ in H&R-algorithm is done by a complicated randomized subroutine which can be sketched as follows: Because of P_2, we only need to consider how to find a triangle packing of large weight when $\mathcal{O}pt$ contains a heavy set of edges between cycles in $\mathcal{C}$. So, assume that $\mathcal{O}pt$ contains a heavy set of edges between cycles in $\mathcal{C}$. Then, G must contain a heavy matching M consisting of edges whose endpoints belong to different cycles in $\mathcal{C}$. Thus, we want P_3 to contain not only as many edges of $\mathcal{C}$ but also as many edges of M as possible. Towards this goal, one way is to mark each edge of $\mathcal{C}$ with a probability $0 < q < 1$ such that adjacent edges in $\mathcal{C}$ and edges in different cycles in $\mathcal{C}$ are marked independently at random and each cycle in $\mathcal{C}$ has at least one edge marked. Let R be the set of marked edges. We say that an edge e of M *survives* the marking process if both endpoints of e become of degree at most 1 in graph $\mathcal{C} - R$. Note that each edge of M survives the marking process with probability $(2q - q^2)^2$. Let $\mathcal{C}'$ be the graph obtained from $\mathcal{C} - R$ by adding the edges of M. Then, the expected weight of $\mathcal{C}'$ is $(1-q)w(\mathcal{C}) + (2q - q^2)^2 w(M)$. Moreover, each connected component of $\mathcal{C}'$ is either a path, or a cycle containing at least two edges of M. So, after removing one edge of M from each cycle in $\mathcal{C}'$, we obtain a collection of paths whose expected total weight is at least $(1-q)w(\mathcal{C}) + \frac{1}{2}(2q - q^2)^2 w(M)$. These paths are then patched together into a Hamiltonian cycle of G which is then cut into paths of length 2 by removing one third of its edges. The resulting paths of length 2 lead to a triangle packing whose expected weight is at least $\frac{2}{3}(1-q)w(\mathcal{C}) + \frac{1}{3}(2q-q^2)^2 w(M)$, which is large if $w(M)$ is large.

4 New Computation of P_3

This section details our new computation of P_3, which is basically a significant refinement of the computation of P_3 in H&R-algorithm. We inherit the notations in the last section.

There are two main ideas in our new computation of P_3. First, the weight of the matching M computed in H&R-algorithm (outlined above) may be as small as $\frac{1}{3}w(\mathcal{O}pt)$. In our new computation, we instead compute a maximum-weight

b-matching M_1 consisting of edges whose endpoints belong to different cycles in $\mathcal{C}$, where $b(v) = 2$ for every vertex v. Note that $w(M_1)$ is close to $w(\mathcal{O}pt)$ when $\mathcal{O}pt$ contains a heavy set of edges between cycles in $\mathcal{C}$. If we are lucky enough that M_1 contains no cycle of odd length, then we can partition M_1 into two matchings among which the heavier one has weight close to $\frac{1}{2}w(\mathcal{O}pt)$ and we can use it as M. Suppose that C is a cycle of odd length in M_1. The crucial point is that with a significantly high probability, at least one edge of C will not survive the marking process. Thus, with a significantly high probability, we can partition the survived edges of M_1 into two matchings among which the heavier one can be used as M.

Second, it is unnecessary to require that each triangle C in $\mathcal{C}$ have at least one edge marked because if C has no edge marked then C can be included in P_3 as it is. That is, we may distinguish the triangles in $\mathcal{C}$ from the other cycles and mark the edges in triangles in $\mathcal{C}$ with a smaller probability (so that the edges in triangles have a larger probability to remain in P_3).

Next, we detail our new computation of P_3 from $\mathcal{C}$. The first step is 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 $v \in V(G_1)$. (*Comment:* Note that $b(v)$ is an upper bound on the degree of v in G_1 and hence is not necessarily the degree of v in G_1.)

Note that $w(M_1)$ is close to $w(\mathcal{O}pt)$ when $\mathcal{O}pt$ contains a heavy set of edges between cycles in $\mathcal{C}$. So, we want to add the edges of M_1 to $\mathcal{C}$. However, adding the edges of M_1 to $\mathcal{C}$ yields a graph which may have a lot of vertices of degree 3 or 4 and is hence far from a triangle packing of G. To remedy this situation, we want to compute a set R of edges in $\mathcal{C}$ and a subset M of M_1 such that adding the edges of M to $\mathcal{C} - R$ yields a graph in which each connected component is a cycle or path.

The next two steps of our algorithm are for computing the set R. Before describing the details, we need to define several notations. Let $C_1, \ldots, C_r$ be the cycles in $\mathcal{C}$. Moreover, throughout the remainder of this section, 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 11. Note that $0.276 < p < 0.277$; hence $(1-p)^2 > \frac{1}{2}$ and $\frac{27}{80}p - \frac{9}{80}p^2 \geq \frac{27}{320}$. Now, we are ready to describe Steps 2 and 3 of our algorithm.

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.
   (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 $c \equiv 3 \pmod 4$, 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 $R = R_1 \cup \cdots \cup R_r$.

The next lemma is obvious from Step 2b:

Lemma 3. *For every triangle C_i in $\mathcal{C}$ and for every vertex v of C_i, the following hold:*

1. *v is incident to no edge of R with probability $(1-p)^2$.*
2. *v is incident to exactly one edge of R with probability $2p(1-p)$.*
3. *v is incident to exactly two edges of R with probability p^2.*

Note that our algorithm processes those cycles C_i of $\mathcal{C}$ with $|C_i| \geq 4$ as in the H&R-algorithm. So, we have the following lemma:

Lemma 4. *For every cycle C_i of $\mathcal{C}$ with $|C_i| \geq 4$, the following hold:*

1. *For every edge e of C_i, $\Pr[e \in R] = \frac{1}{4}$.*
2. *For every vertex v of C_i, v is incident to at least one edge of R with probability $\frac{1}{2}$.*

Proof. We include the proof for completeness. Consider an arbitrary cycle C_i of $\mathcal{C}$ with $|C_i| \geq 4$. By the comments on Steps 2(c)iv through 2(c)vi, we have $\mathcal{E}[|R_i|] = \frac{|C_i|}{4}$. Moreover, each edge of C_i is added to R_i with the same probability. Thus, $\Pr[e \in R_i] = \frac{1}{4}$ 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 $\frac{1}{2}$. □

We next turn to the computation of the subset M of M_1. Steps 4 through 9 of our algorithm are for this purpose.

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:* $M \subseteq M_1$ is a matching of G.)
9. Let $\mathcal{C}'$ be the graph obtained from graph $\mathcal{C} - R$ by adding the edges in M.

The next lemma is clear from the above construction of $\mathcal{C}'$:

Lemma 5. *The following hold:*

1. *Each connected component of $\mathcal{C}'$ is a cycle or path.*
2. *Every triangle in $\mathcal{C}'$ is also a triangle in $\mathcal{C}$.*
3. *Every cycle C in $\mathcal{C}'$ with $|C| \geq 4$ contains at least two edges in M.*

The next lemma will be used to show that for each edge $e \in M_1$, e is included in the output triangle packing by our algorithm with high probability.

Lemma 6. *For each $e \in M_1$, $\Pr[e \in M \mid e \in M_2] \geq \frac{9}{20}$.*

Proof. Assume that the event $e = \{u, v\} \in M_2$ occurs. Let K be the connected component of G_2 that contains e. If K is not an odd cycle, then clearly e is in M with probability at least $\frac{1}{2} \geq \frac{9}{20}$. So, assume that K is an odd cycle. We distinguish two cases as follows:

Case 1: K is an odd cycle of length at least 5. In this case, K must contain a vertex $z \notin \{u, v\}$ such that the cycle in $\mathcal{C}$ containing z contains neither u nor v. Let B_z be the event that the degree of z in the graph $\mathcal{C} - R$ is 2. By Statement 1 of Lemma 3 and Statement 2 of Lemma 4, B_z occurs with probability $(1-p)^2$ or $\frac{1}{2}$ independently of the event $e \in M_2$. Obviously, when B_z occurs, e is contained in M with probability at least $\frac{1}{2}$. On the other hand, when B_z does not occur, e is contained in M with probability at least $\frac{|K|-1}{|K|} \cdot \frac{1}{2} \geq \frac{2}{5}$. So, the probability that e is in M is at least $\frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{2}{5} = \frac{9}{20}$.

Case 2: K is a triangle. In this case, let z be the vertex in K other than u and v. Note that u, v, and z belong to different cycles in $\mathcal{C}$. Let B_z be the event that the degree of z in the graph $\mathcal{C} - R$ is 2. By Statement 1 of Lemma 3 and Statement 2 of Lemma 4, B_z occurs with probability $(1-p)^2$ or $\frac{1}{2}$ independently of the event $e \in M_2$. Obviously, when B_z occurs, e is definitely contained in M because of Step 7. On the other hand, when B_z does not occur, e is contained in M with probability at least $\frac{1}{3}$. So, the probability that e is in M is at least $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{3} = \frac{2}{3} > \frac{9}{20}$. □

By Lemma 5, $\mathcal{C}'$ is a collection of disjoint paths and cycles. Our plan is to transfrom $\mathcal{C}'$ into a triangle packing of G as follows:

- First, break each nontriangle cycle in $\mathcal{C}'$ by removing one edge.
- Next, patch the path components of $\mathcal{C}'$ together into a single path Y.
- Finally, cut Y into paths of length 2 by removing one third of its edges. (*Comment:* Each path of length 2 can be trivially transformed into a triangle by adding the edge between its endpoints.)

The nontriangle cycles in $\mathcal{C}'$ should be broken carefully. Steps 10 through 11 of our algorithm are for this purpose.

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 to obtain two paths Q_1 and Q_2 of length 2, add the edge between the endpoints of Q_1, and add the edge between the endpoints of Q_2. (*Comment:* Because of the maximality of $\mathcal{C}$, this step does not decrease $w(\mathcal{C}')$. Moreover, after this step, each cycle of $\mathcal{C}'$ is a triangle.)

We next show that each edge of M_1 remains in $\mathcal{C}'$ after Step 11 with high probability.

Lemma 7. *For each edge $e \in M$ such that at least one endpoint of e appears in a non-triangle in $\mathcal{C}$, e survives the deletion in Step 11 with probability at least $\frac{3}{4}$.*

Proof. Without loss of generality, we may assume that C_1 is not a triangle and contains one endpoint v of e. Because the cycles in $\mathcal{C}$ are processed independently in Step 2, we may assume that C_1 is processed after the other cycles in $\mathcal{C}$ have been processed.

Consider the time point t at which our algorithm has just finished processing all the cycles in $\mathcal{C}$ other than C_1. Let $\mathcal{S}^t$ be the set of all matchings N in the graph $(V(G), M_1)$ such that each connected component of the graph obtained from $\mathcal{C} - (E(C_1) \cup R_2 \cup \cdots \cup R_r)$ by adding the edges of N is a path or cycle. For each matching $N \in \mathcal{S}^t$, let $p^t(N)$ be the probability that the matching M constructed in Step 8 equals N. Note that $p^t(N)$ only depends on the random choices made by our algorithm when processing C_1 and later in Steps 5 and 8. Further let B_e^t be the event that e is contained in $\mathcal{C}'$ immediately after Step 11.

Let $\mathcal{S}_e^t$ be the set of all matchings $N \in \mathcal{S}^t$ with $e \in N$ and $p^t(N) > 0$. We claim that for each matching $N \in \mathcal{S}_e^t$, $\Pr[B_e^t \mid M = N] \geq \frac{3}{4}$. If this claim indeed holds, then

$$\Pr[B_e^t \mid e \in M] = \sum_{N \in \mathcal{S}_e^t} \Pr[B_e^t \mid M = N] \cdot \frac{p^t(N)}{\sum_{N' \in \mathcal{S}_e^t} p^t(N')} \geq \frac{3}{4}$$

which implies the lemma immediately. So, it remains to prove the claim.

To prove the claim, consider an arbitrary matching $N \in \mathcal{S}_e^t$. Assume that the event $M = N$ occurs. Let K be the graph obtained from $\mathcal{C} - (E(C_1) \cup R_2 \cup \cdots \cup R_r)$ by adding the edges of N. Let Q be the connected component of K in which v appears. Note that Q is a path and v is an endpoint of Q. Let u be the endpoint of Q other than v. If u does not appear in C_1, then B_e^t occurs with probability 1.

So, assume that u appears in C_1. Then, because $|C_1| \geq 4$ and our new algorithm processes each non-triangle C_i in $\mathcal{C}$ in the same way as H&R-algorithm does, Lemma 2 in [3] guarantees that B_e^t occurs with probability at least $\frac{3}{4}$. This completes the proof of the claim and hence that of the lemma. □

Lemma 8. *For each $e \in M_1$ such that neither endpoint of e appears in a triangle in $\mathcal{C}$, e is contained in $\mathcal{C}'$ immediately after Step 11 with probability at least $\frac{27}{320}$.*

Proof. Consider an edge $e = \{u, v\}$ in M_1 such that neither u nor v appears in a triangle in $\mathcal{C}$. Let B_u (respectively, B_v) be the event that u (respectively, v) is incident to one edge in R. Note that when both B_u and B_v occur, e is contained in M_2. So, by Lemma 6, $\Pr[e \in M \mid E_u \text{ and } E_v] \geq \frac{9}{20}$. Moreover, by the comment on Step 3, $\Pr[E_u \text{ and } E_v] = \frac{1}{4}$. Hence, $\Pr[e \in M] \geq \frac{9}{80}$. Consequently, by Lemma 7, the probability that e is contained in $\mathcal{C}'$ immediately after Step 11 is at least $\frac{9}{80} \cdot \frac{3}{4} = \frac{27}{320}$. □

Lemma 9. *For each $e \in M_1$ such that exactly one endpoint of e appear in a triangle in $\mathcal{C}$, e is contained in $\mathcal{C}'$ immediately after Step 11 with probability at least $\frac{27}{320}$.*

Proof. Consider an edge $e = \{u, v\}$ in M_1 such that u appears in a triangle in $\mathcal{C}$ but v does not. Let B_1 be the event that u is incident to exactly one edge in R and so does v. Similarly, let B_2 be the event that u is incident to exactly two edges in R and v is incident to exactly one edge in R. Note that when B_1 or B_2 occurs, e is contained in M_2. So, by Lemma 6, $\Pr[e \in M \mid E_1] \geq \frac{9}{20}$ and $\Pr[e \in M \mid E_2] \geq \frac{9}{20}$. Moreover, by Statement 2 of Lemma 3 and Statement 2 of Lemma 4, $\Pr[E_1] = 2p(1-p) \cdot \frac{1}{2} = p(1-p)$ and $\Pr[E_2] = p^2 \cdot \frac{1}{2} = \frac{1}{2}p^2$. Hence, $\Pr[e \in M \text{ and } E_1] \geq \frac{9}{20}p(1-p)$ and $\Pr[e \in M \text{ and } E_2] \geq \frac{9}{40}p^2$.

Obviously, when B_1 occurs and $e \in M$, e survives the deletion in Step 11 with probability at least $\frac{3}{4}$ by Lemma 7. The crucial point is that when B_2 occurs and $e \in M$, e survives the deletion in Step 11 with probability 1. Therefore, the probability that e is contained in $\mathcal{C}'$ immediately after Step 11 is at least $\frac{9}{20}p(1-p) \cdot \frac{3}{4} + \frac{9}{40}p^2 \cdot 1 \geq \frac{27}{320}$. □

Lemma 10. *Suppose that $e = \{u_1, v_1\}$ is an edge in M such that both u_1 and v_1 appear in triangles 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 $\frac{3}{4}$.*

Proof. Without loss of generality, we may assume that C_1 (respectively, C_2) is the cycle in $\mathcal{C}$ to which u_1 (respectively, v_1) belongs. Because the cycles in $\mathcal{C}$ are processed independently in Step 2, we may assume that C_1 and C_2 are processed after the other cycles in $\mathcal{C}$ have been processed.

Consider the time point t at which our algorithm has just finished processing all the cycles in $\mathcal{C}$ other than C_1 and C_2. Let $\mathcal{S}^t$ be the set of all matchings N in the graph $(V(G), M_1)$ such that each connected component of the graph obtained from $\mathcal{C} - (E(C_1) \cup E(C_2) \cup R_3 \cup \cdots \cup R_r)$ by adding the edges of N is a path or cycle. For each matching $N \in \mathcal{S}^t$, let $p^t(N)$ be the probability that the matching M constructed in Step 8 equals N. Note that $p^t(N)$ only depends on the random choices made by our algorithm when processing C_1 and C_2 and later in Steps 5 and 8. Further let $B^t_{1,1}$ be the event that both u_1 and v_1 are incident to exactly one edge in R, and let B^t_e be the event that e is contained in $\mathcal{C}'$ immediately after Step 11.

Let $\mathcal{S}^t_e$ be the set of all matchings $N \in \mathcal{S}^t$ with $e \in N$ and $p^t(N) > 0$. We claim that for each matching $N \in \mathcal{S}^t_e$, $\Pr[B^t_e \mid M = N \text{ and } B^t_{1,1}] \geq \frac{3}{4}$. If this claim indeed holds, then

$$\Pr[B^t_e \mid e \in M \text{ and } B^t_{1,1}] = \sum_{N \in \mathcal{S}^t_e} \Pr[B^t_e \mid M = N \text{ and } B^t_{1,1}] \cdot \frac{p^t(N)}{\sum_{N' \in \mathcal{S}^t_e} p^t(N')} \geq \frac{3}{4}$$

which implies the lemma immediately. So, it remains to prove the claim.

To prove the claim, consider an arbitrary matching $N \in \mathcal{S}^t_e$. Assume that the events $M = N$ and $B^t_{1,1}$ occur. Then, B^t_e occurs with probability at least $\frac{1}{2}$ because each cycle C in the graph $\mathcal{C}'$ constructed in Step 9 with $E(C) \cap M_1 \neq \emptyset$ contains at least two edges of N. Our goal is to show that B^t_e occurs with probability at least $\frac{3}{4}$. First, we need several definitions:

– Let u_2 and u_3 be the vertices of C_1 other than u_1.
– Let v_2 and v_3 be the vertices of C_2 other than v_1.
– Let H^t be the graph obtained from $\mathcal{C} - (E(C_1) \cup E(C_2) \cup R_3 \cup \cdots \cup R_r)$ by adding the edges of N. (*Comment:* The degree of each vertex of $V(C_1) \cup V(C_2)$ in H^t is 0 or 1.)

Since N is a matching, there are at most two paths Q in H^t such that one endpoint of Q is in $\{u_2, u_3\}$ and the other is in $\{v_2, v_3\}$. We distinguish three cases as follows:

Case 1: There is no path Q in H^t such that one endpoint of Q is in $\{u_2, u_3\}$ and the other is in $\{v_2, v_3\}$. In this case, it is easy to see that B^t_e occurs with probability 1.

Case 2: There are two paths Q in H^t such that one endpoint of Q is in $\{u_2, u_3\}$ and the other is in $\{v_2, v_3\}$. Without loss of generality, we may assume that H^t contains a path between u_2 and v_2 and contains another path between u_3 and v_3. Note that each vertex in $V(C_1) \cap V(C_2)$ is incident to an edge in N. So, exactly one of the following four events occurs:

– $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_3\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
– $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_2\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
– $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
– $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_3\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.

Obviously, when either of the first two events occurs, e does not appear in a cycle in the graph $\mathcal{C}'$ constructed in Step 9 and hence B^t_e occurs with probability 1. Moreover, if we concentrate on the random choices made when processing C_1 and C_2, the four events occur with the same probability (namely, $(1-p)^2p^4$). Thus, B^t_e occurs with probability at least $\frac{2(1-p)^2p^4}{4(1-p)^2p^4} \cdot 1 + \frac{2(1-p)^2p^4}{4(1-p)^2p^4} \cdot \frac{1}{2} = \frac{3}{4}$.

Case 3: There is exactly one path Q in H^t such that one endpoint of Q is in $\{u_2, u_3\}$ and the other is in $\{v_2, v_3\}$. Without loss of generality, we may assume that H^t contains a path between u_2 and v_2. If N contains an edge incident to u_3 and another edge incident to v_3, then the same discussion in Case 2 applies. So, it suffices to consider the following two subcases:

Case 3.1: Exactly one of u_3 and v_3 is incident to an edge in N. Without loss of generality, we may assume that u_3 is incident to an edge in N but v_3 is not. Then, each vertex in $V(C_1) \cap V(C_2)$ other than v_3 is incident to an edge in N. So, exactly one of the following six events occurs:

- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}, \{u_2, u_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_3\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_2\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_2\}, \{u_2, u_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_3\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.

Obviously, when either of the first four events occurs, e does not appear in a cycle in the graph $\mathcal{C}'$ constructed in Step 9 and hence B_e^t occurs with probability 1. Moreover, if we concentrate on the random choices made when processing C_1 and C_2, then the first and the fifth events occur with the same probability (namely, $(1-p)^3p^3$), while the other events occur with the same probability (namely, $(1-p)^2p^4$). Thus, B_e^t occurs with probability at least $\frac{(1-p)^3p^3+3(1-p)^2p^4}{2(1-p)^3p^3+4(1-p)^2p^4} \cdot 1 + \frac{(1-p)^3p^3+(1-p)^2p^4}{2(1-p)^3p^3+4(1-p)^2p^4} \cdot \frac{1}{2} \geq \frac{3}{4}$.

Case 3.2: Neither u_3 nor v_3 is incident to an edge in N. In this case, each vertex in $V(C_1) \cap V(C_2)$ other than u_3 and v_3 is incident to an edge in N. So, exactly one of the following nine events occurs:

- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}, \{u_2, u_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_3\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_2\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_2\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_2\}, \{v_1, v_3\}, \{v_2, v_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_2\}, \{u_2, u_3\}\}$.
- $R \cap (E(C_1) \cup E(C_2)) = \{\{u_1, u_3\}, \{v_1, v_3\}, \{u_2, u_3\}, \{v_2, v_3\}\}$.

Obviously, when one of the first five events occurs, e does not appear in a cycle in the graph $\mathcal{C}'$ constructed in Step 9 and hence B_e^t occurs with probability 1. As for the last four events, we can say the following:

- Suppose that $|E(Q)| = 1$. Then, when the sixth event occurs, e appears in a good cycle in the graph $\mathcal{C}'$ constructed in Step 9 and hence B_e^t occurs with probability 1. Of course, when the ith event with $i \in \{7, 8, 9\}$ occurs, e appears in an ordinary cycle K in the graph $\mathcal{C}'$ constructed in Step 9 with $|E(K) \cap N| \geq 2$ and hence B_e^t occurs with probability $\frac{1}{2}$.

– Suppose that $|E(Q)| \geq 2$. Then, Q contains at least two edges of N. So, when the ith event with $i \in \{6, \ldots, 9\}$ occurs, e appears in an ordinary cycle K with $|E(K) \cap N| \geq 3$ in the graph $\mathcal{C}'$ constructed in Step 9 and hence B^t_e occurs with probability at least $\frac{2}{3}$.

Furthermore, if we concentrate on the random choices made when processing C_1 and C_2, the first, the second, the seventh, and the eighth events occur with the same probability (namely, $p^3(1-p)^3$), the third, the fourth, the fifth, and the ninth events occur with the same probability (namely, $p^4(1-p)^2$), and the sixth event occurs with probability $p^2(1-p)^4$. Thus, if $|E(Q)| = 1$, then B^t_e occurs with probability at least $\frac{2p^3(1-p)^3+3p^4(1-p)^2+p^2(1-p)^4}{4p^3(1-p)^3+4p^4(1-p)^2+p^2(1-p)^4} \cdot 1 + \frac{2p^3(1-p)^3+p^4(1-p)^2}{4p^3(1-p)^3+4p^4(1-p)^2+p^2(1-p)^4} \cdot \frac{1}{2} \geq \frac{3}{4}$. On the other hand, if $|E(Q)| \geq 2$, then B^t_e occurs with probability at least $\frac{2p^3(1-p)^3+3p^4(1-p)^2}{4p^3(1-p)^3+4p^4(1-p)^2+p^2(1-p)^4} \cdot 1 + \frac{p^2(1-p)^4+2p^3(1-p)^3+p^4(1-p)^2}{4p^3(1-p)^3+4p^4(1-p)^2+p^2(1-p)^4} \cdot \frac{2}{3} \geq \frac{3}{4}$. This completes the proof of the claim and hence that of the lemma. □

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

Proof. Consider an edge $e = \{u, v\}$ in M_1 such that both u and v appear in a triangle in $\mathcal{C}$. Let B_0 be the event that both u and v are incident to exactly one edge in R, and let B_1 be the event that both u and v are incident to at least one edge in R and at least one of them is incident to exactly two edges in R.

Note that when B_0 or B_1 occurs, e is contained in M_2. So, by Lemma 6, $\Pr[e \in M \mid E_0] \geq \frac{9}{20}$ and $\Pr[e \in M \mid E_1] \geq \frac{9}{20}$. Moreover, by Statements 2 and 3 of Lemma 3 $\Pr[E_0] = 2p(1-p) \cdot 2p(1-p) = 4p^2(1-p)^2$ and $\Pr[E_1] = p^2 \cdot 2p(1-p) + 2p(1-p) \cdot p^2 + p^4 = 4p^3 - 3p^4$. Hence, $\Pr[e \in M \text{ and } E_0] \geq \frac{9}{5}p^2(1-p)^2$ and $\Pr[e \in M \text{ and } E_1] \geq \frac{9}{20}(4p^3 - 3p^4)$.

By Lemma 10, when B_0 occurs and $e \in M$, e survives the deletion in Step 11 with probability at least $\frac{3}{4}$. The crucial point is that when B_1 occurs and $e \in M$, e survives the deletion in Step 11 with probability 1. Therefore, the probability that e is contained in $\mathcal{C}'$ immediately after Step 11 is at least $\frac{9}{5}p^2(1-p)^2 \cdot \frac{3}{4} + \frac{9}{20}(4p^3 - 3p^4) \cdot 1 = \frac{27}{20}p^2 - \frac{9}{10}p^3 \geq \frac{27}{320}$. □

By the comment on Step 12, each connected component of $\mathcal{C}'$ after Step 12 is a triangle or path. So, it is now almost trivial to transform $\mathcal{C}'$ into a triangle packing. Step 12 of our algorithm is for this purpose.

12. If $\mathcal{C}'$ has at least one path component, then perform the following steps:
    (a) Connect the path components of $\mathcal{C}'$ into a single cycle Y by adding some edges of G.
    (b) Break Y into paths each of length 2 by removing a set F of edges from Y with $w(F) \leq \frac{1}{3} \cdot w(Y)$.
    (c) For each path Q of length 2 obtained in Step 12b, add the edge between the endpoints of Q.

    (*Comment:* After this step, $\mathcal{C}'$ is a triangle packing of G.)
13. Let $P_3 = \mathcal{C}'$.

5 Analysis of the Approximation Ratio

By the comment on Step 3, the expected total weight of edges of $\mathcal{C}$ remaining in $\mathcal{C}'$ immediately after Step 11 is at least $\left((1-p)\alpha + \frac{3}{4}(1-\alpha)\right) w(\mathcal{C}) = \left(\frac{3}{4} - (p-\frac{1}{4})\alpha\right) w(\mathcal{C}) \geq \left(\frac{3}{4} - (p-\frac{1}{4})\alpha\right)(1-\epsilon)w(\mathcal{O}pt)$. Moreover, by Lemmas 8, 9, and 11, the expected total weight of edges of M_1 remaining in $\mathcal{C}'$ immediately after Step 11 is at least $\frac{27}{320}w(M_1)$. Furthermore, by the construction of M_1, $w(M_1)$ is larger than or equal to the total weight of those edges $\{u,v\}$ such that some triangle in $\mathcal{O}pt$ contains both u and v but no cycle in $\mathcal{C}$ contains both u and v. So, $w(M_1) \geq (1-\beta)w(\mathcal{O}pt)$. Now, since $w(P_3)$ is at least two-thirds of the total weight of edges in $\mathcal{C}'$ immediately after Step 11, we have

$$\mathcal{E}[w(P_3)] \geq \frac{2}{3}\left(\frac{3}{4} - (p-\frac{1}{4})\alpha\right)(1-\epsilon)w(\mathcal{O}pt) + \frac{2}{3}\cdot\frac{27}{320}(1-\beta)w(\mathcal{O}pt) \quad (1)$$

$$= \left(\frac{89}{160} - \frac{1}{2}\epsilon - \frac{2}{3}(p-\frac{1}{4})(1-\epsilon)\alpha - \frac{9}{160}\beta\right) w(\mathcal{O}pt). \quad (2)$$

So, by Lemmas 1 and 2, we have

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

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

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

In summary, we have proven the following theorem:

Theorem 1. *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. Chlebík, M., Chlebíková, J.: Approximating Hardness for Small Occurrence Instances of NP-Hard Problems. In: Petreschi, R., Persiano, G., Silvestri, R. (eds.) CIAC 2003. LNCS, vol. 2653, pp. 152–164. Springer, Heidelberg (2003)
2. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
3. Hassin, R., Rubinstein, S.: An Approximation Algorithm for Maximum Triangle Packing. Discrete Applied Mathematics 154, 971–979 (2006)
4. Hassin, R., Rubinstein, S.: Erratum to gAn approximation algorithm for maximum triangle packingh: [Discrete Applied Mathematics 154 (2006) 971-979]. Discrete Applied Mathematics 154, 2620–2620 (2006)
5. Kann, V.: Maximum Bounded 3-Dimensional Matching Is MAX SNP-Complete. Information Processing Letters 37, 27–35 (1991)
6. Raghavan, P.: Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. Journal of Computer and System Sciences 38, 683–707 (1994)