On Random Greedy Triangle Packing

David A. Grable*
Institut für Informatik
Humboldt-Universität zu Berlin
D-10099 Berlin Germany

Submitted: August 5, 1996; Accepted: February 26, 1997

Abstract

The behaviour of the random greedy algorithm for constructing a maximal packing of edge-disjoint triangles on n points (a maximal partial triple system) is analysed with particular emphasis on the final number of unused edges. It is shown that this number is at most $n^{7/4+o(1)}$, "halfway" from the previous best-known upper bound $o(n^2)$ to the conjectured value $n^{3/2+o(1)}$. The more general problem of random greedy packing in hypergraphs is also considered.

1 Introduction

Consider the following simple algorithm for constructing a maximal collection of pair-disjoint triples in a set of n points: repeatedly pick a triple uniformly at random from among all triples which do not share a pair with any previously picked triple, until there are no more candidate triples. It is perhaps mildly surprising that such a simple random greedy procedure almost always results in a collection of triples which cover almost all of the pairs [13, 12]. In this paper we obtain significantly tighter bounds on the number of uncovered pairs. In particular, we show that the number of uncovered pairs is almost always no more than $n^{7/4+o(1)}$, where $o(1)$ is a function going to 0 as n goes to infinity.

This problem is expressed nicely in the language of design theory. A *partial triple system* on n points (a PTS(n) for short) is a collection of 3-element subsets (*triples*) of $\{1,\ldots,n\}$ such that each 2-element subset (*pair*) is contained in (*covered by*) at most one triple. Of considerable interest are partial triple systems in which every pair is covered by exactly one triple. Such systems are called *Steiner triple systems*. The reader is referred to [3] for more background on design theory.

A partial triple system is *maximal* if no triple can be added without covering some pair more than once. It is obvious, but worth noting, that Steiner triple systems are maximal, but they are

*Supported by Deutsche Forschungsgemeinschaft project number Pr 296/4-1.
e-mail: grable@informatik.hu-berlin.de
Mathematics Subject Classification: 05B40, 05C70

not the only maximal partial triple systems. Another observation is that the *leave graph* of a PTS (the graph whose edges are the uncovered pairs) is triangle-free if and only if the PTS is maximal.

The random greedy algorithm constructs a maximal partial triple system in the following way. It starts with an empty partial triple system and the complete list of *candidate* triples. It repeatedly picks a candidate triple from the list uniformly at random, adds it to the partial triple system, and removes it and all other candidates with which it shares a pair from the list of candidates, until there are no more candidates.

The problem is to determine how many uncovered pairs remain. In other words, we're interested in finding out how close the resulting partial triple system is to a Steiner triple system. Of course, since the procedure involves randomness, it may be that with some small probability, the resulting PTS is very bad. On the other hand, there is a small probability that the result is a Steiner triple system. It will turn out, though, that usually the result is somewhere in between, but not very far from being a Steiner triple system.

There are two alternate ways to express the algorithm. One, from the point of view of the leave graph, is that the algorithm starts with the complete graph and repeatedly removes random triangles until the graph is triangle-free.

In the other alternate expression, all of the randomness takes place in the initial stage of execution: the list of all triples is first randomly ordered and then the triples are considered one at a time in this order. If the triple shares no pair with any previously selected triple (i.e. it can be added to the PTS) then it is added. Otherwise, it is discarded. It is actually this last procedure which we will analyse.

We take what we call an "honest nibble" approach. The nibble method, pioneered by Rödl [11], was originally a method (really an algorithm) for showing the existence of asymptotically good partial designs as conjectured by Erdős and Hanani [4] and, later, the existence of packings and colourings in hypergraphs [5, 10, 7, 1, 9]. The nibble algorithm isn't exactly the same as the random greedy algorithm but there are enough similarities that Rödl and Thoma [12] were able to use it to show that the random greedy algorithm almost always produces a partial triple system with only $o(n^2)$ uncovered pairs. Spencer [13] proved the same using completely different techniques. By "almost always" we mean that with probability $1 - o(1)$, the algorithm produces the required result.

We modify the nibble algorithm so that it behaves exactly the same as the random greedy algorithm. This "honest nibble" algorithm is presented in the next section. We then give a simplified analysis that shows that the number of uncovered pairs is almost always at most $n^{11/6+o(1)}$. Thereafter, we will show how to refine the analysis to improve the bound to at most $n^{7/4+o(1)}$.

No lower bounds on the number of uncovered pairs are known, but substantial computer simulations of the random greedy algorithm have been carried out by Balińska and Wieczorek [2] for triangle packing and by Gordon, Kuperberg, Patashnik, and Spencer [6] for more general packing problems. These simulations indicate that the correct order of magnitude for the number of uncovered pairs is $n^{3/2+o(1)}$ and, indeed, Joel Spencer has offered \$200 for a proof.

At the end of the paper, we will also look at the behaviour of the random greedy algorithm for constructing packings in hypergraphs.

2 The Honest Nibble

The traditional nibble process starts with the empty partial triple system and repeatedly adds a small number of triples in the following way: triples are selected independently at random with a very small probability from all remaining candidates, each selected triple which does not share a pair with any other selected triple is added to the current partial design, and then the list of candidates is updated by removing all selected triples and any triples with which they share a pair. To get the desired theoretical result, it is only necessary to take a large but fixed number of such nibbles, but one could of course run this process until there were no more candidate triples.

To make the connection to the random greedy algorithm, consider the version of the random greedy algorithm where all possible triples are first randomly ordered and then the triples are considered one by one. Equivalently, we could generate the ordering in *bursts*. Each of these bursts corresponds roughly to a single nibble.

More specifically, do the following. Start with an empty partial triple system and the set of all triples. To generate a burst, select triples independently at random with some small probability (to be specified later), randomly order the selected triples, and remove the selected triples from the set of unused triples. After generating each burst, process the triples in the manner of the greedy algorithm: consider each triple in turn in the random order given and, if it can be added to the partial triple system, do so.

Compare what happens here with what happens in the traditional nibble process. Here, if a selected triple could not be added to the partial triple system as it existed at the beginning of the burst, it will never be added. In the traditional nibble, such a triple would never have been selected since we only select from candidate triples—that is, from triples which could be added. Here, of the remaining selected triples, a triple which is pair-disjoint from all other remaining selected triples will always be placed in the partial design. The same thing happens in the traditional nibble. It is only in the case of selected triples which share a pair that the two methods differ. The traditional nibble throws out all such triples, but here we "do the right thing", by considering the ordering and doing what the greedy algorithm would do. Note that this is the only situation where the ordering of the burst matters.

The point of the bursts is to give us well defined intermediate stopping points in the process, where we can determine what the leave graph of the partial triple system looks like. This way, we only have to analyse one burst at a time, giving more or less a proof by induction. By taking the right sized burst, we hope to show that the leave graph remains fairly regular throughout the entire process. This will let us predict how many edges are left near the end fairly well. If we took bursts that were too small—in the extreme case, only one triangle at a time—we couldn't take much advantage of the fact that the triangles are presented randomly. The burst would only affect a very few vertices, so the degrees wouldn't stay very regular from one burst to the next. If we took very large bursts—the extreme case here is a single burst—the ordering dependencies among those selected triangles sharing pairs would be very difficult to analyse. We therefore pick bursts of a size such that they each contain enough randomly placed triangles that their effects occur in a smooth way over the entire leave graph, but not so many triangles that there are a lot of dependencies.

We are interested in showing that at the end, when the leave graph of the partial triple system is triangle-free, there are almost always fewer than $f(n)$ edges, for some function $f(n)$. We'll do that by carefully analysing the leave graph after each burst and showing that there is a number t so that after the t^{th} burst, the leave graph already has fewer than $f(n)$ edges.

At last we introduce some notation. Let G_s be the leave graph of the partial design after the s^{th} burst. So

$$K_n = G_0 \supseteq G_1 \supseteq \cdots \supseteq G_s \supseteq \cdots \supseteq G_t \supseteq \cdots .$$

The probability with which we choose triangles in the s^{th} burst is the ratio of two quantities: a parameter η to be defined a bit later as a function of n going to 0 as n goes to infinity and a number D_{s-1} defined as follows.

Consider an ideal situation in which every edge of a graph $G \subseteq K_n$ is contained in exactly D triangles. Nibble in G with probability η/D (that is, select triangles independently at random with probability η/D, order the selected nibbles at random, etc.) and let H be the probability that a given edge e is not removed from the graph. This probability turns out, as we will later show, to be independent of the choice of e and even of G. We will see that, indeed,

$$H := \Pr[\text{edge survival}] = 1 - \eta + \tfrac{3}{2}\eta^2 - \tfrac{5}{2}\eta^3 + O(\eta^4).$$

With this in mind, we define real numbers $D_s := H^{2s} n$ and $E_s := H^s \binom{n}{2}$. When going from G_{s-1} to G_s, we will nibble with probability η/D_{s-1}. Nibbling at this rate will ensure that each edge of G_s is in about D_s triangles and that G_s has about E_s edges.

Just for reference, in the traditional nibble, η is a small constant and $e^{-\eta}$ is used in place of H Indeed, $H = e^{-\eta} + O(\eta^2)$, so the difference in relatively insignificant, but this minor difference benefits us in a major way.

Of course, the D_s and E_s don't track the numbers of triangles contained in each edge and the number of edges exactly. Fix a small positive constant γ and define quantities

$$0 = \varepsilon_0 \le \varepsilon_1 \le \cdots \le \varepsilon_s \le \cdots \le \varepsilon_t \le \gamma$$

such that for each $0 \le s \le t$ and edge $e \in G_s$,

$$\deg_s(e) = (1 \pm \varepsilon_s) D_s = (1 \pm \varepsilon_s) H^{2s} n,$$

where $\deg_s(e)$ is defined to be the number of triangles of G_s containing edge e (the motivation for this seemingly strange concept of degree is explained in the last section along with the connection to packings in hypergraphs, where the edges here will correspond to points and the triangles to hyperedges) and $(1 \pm \varepsilon_s) D_s$ denotes some value in the interval $[(1+\varepsilon_s)D_s, (1-\varepsilon_s)D_s]$. By forcing $\varepsilon_t \le \gamma$, we are really forcing t to be not too large. Similarly, define

$$0 = \delta_0 \le \delta_1 \le \cdots \le \delta_s \le \cdots \le \delta_t \le \gamma$$

such that for each $0 \le s \le t$,

$$|E(G_s)| = (1 \pm \delta_s) E_s = (1 \pm \delta_s) H^s \binom{n}{2},$$

These definitions tell us nothing about how large the ε_s's and the δ_s's are. In the analyses that follow, our goal will be to find upper bounds on these quantities (as functions of s) and therewith determine the maximum t such that ε_t and δ_t are less than γ. This maximum value of t will then minimize $|E(G_t)| \approx E_t = H^t \binom{n}{2}$. Of course, we can adjust γ and η to suit our convenience.

To simplify some of the computations which we encounter along the way, we will assert now that we will choose η so that

$$t \ll n \qquad \text{and} \qquad \eta D_t \gg 1. \tag{1}$$

After choosing η and determining t we will verify that these assertions are indeed valid.

3 Simplified Analysis

In this and the next section, we give an analysis which leads to a non-optimal result in order to illustrate the basic idea. In the following section, we show how to enhance this analysis to improve the result.

Suppose that we have G_{s-1} and know that $\deg_{s-1}(e) = (1 \pm \varepsilon_{s-1}) D_{s-1}$ for every edge $e \in G_{s-1}$ and that $|E(G_{s-1})| = (1 \pm \delta_{s-1}) E_{s-1}$, where $\varepsilon_{s-1}, \delta_{s-1} \le \gamma < 1$. What can we say about ε_s and δ_s?

To simplify the notation, we'll systematically drop the subscript $s-1$ and replace the subscript s by $'$, i.e. G_{s-1} becomes G, G_s becomes G', and similarly for the other symbols. First we bound

ε' by computing the expectation of $\deg'(e)$ for a surviving edge e and then using a large deviation result to bound the actual value with high probability. Afterwards, we'll do the same for $|E(G')|$ to bound δ'.

To compute the expectation, we'll need a few more definitions and two lemmas. We say two triangles are *adjacent* if they share a common edge (but not merely a common vertex). A *path* is a sequence of adjacent triangles and a *cluster* is a maximal collection of selected triangles in which every two triangles are joined by a path within the collection.

The ordering of the triangles within a cluster determines which of the triangles of that cluster succeed, but the success of the triangles in one cluster is independent of the orderings of the other clusters.

The first lemma says that with high probability, we will never encounter clusters with more than a large, but fixed, number of triangles.

Lemma 1 *With probability $1 - O(n^{-2})$, there are no clusters with more than $m = O(1)$ triangles.*

Proof The probability that all c of the triangles of a potential partial (i.e. not necessarily maximal) c-cluster are selected is $(\eta/D)^c$ and the number of potential partial c-clusters is at most $(ED/3) \cdot 3D \cdot 5D \ldots (2c-1)D < n^2(2c)^c D^c$. Therefore, the expected number of partial c-clusters is at most $(2c)^c \eta^c n^2$.

To guarantee that, with probability $1 - O(n^{-2})$, no cluster with m or more triangles is ever selected, we show that the expected number of partial m-clusters is $O(n^{-2})$ and use Markov's inequality. Let $\xi > 0$ be a fixed real number such that $\eta \leq n^{-\xi}$. Note that we will always pick η so that this is possible; for instance in the $n^{11/6+o(1)}$ result η will be chosen to be $n^{-1/3+\gamma}$. Then set $m = 4/\xi = O(1)$. For this value of m, the expected number of partial m-clusters is at most $(2m)^m n^{-m\xi+2} = O(n^{-2})$. □

Since assertion (1) insists that we pick $t \ll n$, it follows that with probability $1 - O(n)$, there are no clusters with more than $m = O(1)$ triangles in *any* of the t bursts. Therefore, we can safely ignore all large clusters.

The next lemma deals with the probability that a constant number of edges all survive a single burst. It says that their survival is almost independent, regardless of their configuration.

Lemma 2 *For any constant-sized set F of edges of G, the probability that all of the edges of F survive to G' is*

$$1 - \tfrac{\eta}{D} \sum_{f \in F} \deg(f) + O(\eta^2).$$

Note The constant concealed by the O-notation is independent of ε, η, and D, and therefore also of s, but does depend on $|F|$. In point of fact, the constant is at most $3|F| + 4|F|^2$, but the exact value is irrelevant.

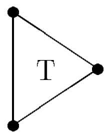
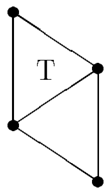
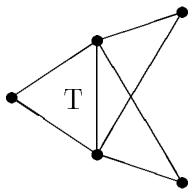
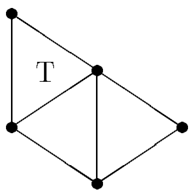
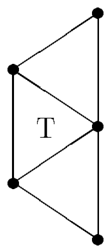
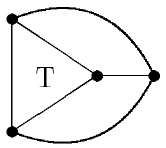
T's cluster	Pr[cluster | T is selected]	Pr[T is accepted | cluster]
	$\left(1-\frac{\eta}{D}\right)^{3D-3} = 1 - 3\eta + \frac{9}{2}\eta^2 - O(\eta^3)$	1
	$3D\frac{\eta}{D}\left(1-\frac{\eta}{D}\right)^{5D-6} = 3\eta - 15\eta^2 + O(\eta^3)$	$\frac{1}{2}$
	$3\binom{D}{2}\left(\frac{\eta}{D}\right)^2\left(1-\frac{\eta}{D}\right)^{7D-9} = \frac{3}{2}\eta^2 + O(\eta^3)$	$\frac{1}{3}$
	$6D^2\left(\frac{\eta}{D}\right)^2\left(1-\frac{\eta}{D}\right)^{7D-9} = 6\eta^2 + O(\eta^3)$	$\frac{2}{3}$
	$3D^2\left(\frac{\eta}{D}\right)^2\left(1-\frac{\eta}{D}\right)^{7D-9} = 3\eta^2 + O(\eta^3)$	$\frac{1}{3}$
	$O(D)\left(\frac{\eta}{D}\right)^2\left(1-\frac{\eta}{D}\right)^{6D-9} = O(\eta^2/D) = O(\eta^3)$	$\frac{1}{3}$
cluster with c triangles	$O(\eta^{c-1})$	$\frac{1}{c} \leq \bullet \leq 1$

Table 1: Small clusters in a graph with $\deg(e) = D$ for every edge e

Proof First compute the probability for one edge e:

$$\Pr[e \notin E'] = \sum_{T \ni e} \Pr[T \text{ is accepted}],$$

where the sum is over all triangles T containing e. Equality follows from the fact that the events "T is accepted" are disjoint.

Fixing T, T is accepted only if it is first selected with probability $\frac{\eta}{D}$. Given that it is selected, it will be accepted depending on what type of cluster it is in:

$$\begin{aligned}\Pr[T \text{ is accepted} \mid T \text{ is selected}] = \sum_{\text{cluster type } C} & \Pr[T \text{ is accepted} \mid \text{cluster type } C] \\ & \times \Pr[\text{cluster type } C \mid T \text{ is selected}].\end{aligned}$$

The small clusters and the relevent probabilities are given in Table 1. In the table we assume that each edge is contained in exactly D triangles; in the present situation each edge is contained in $(1 \pm \varepsilon)D$ triangles, so we must introduce additional $(1 \pm \varepsilon)$ factors. Keep in mind, however, that the probability of selection remains exactly η/D. The first few terms of the sum, corresponding to clusters with 3 or fewer triangles, are

$$\begin{aligned}&[1 - 3(1 \pm \varepsilon)\eta + \tfrac{9}{2}(1 \pm \varepsilon)^2\eta^2 - O(\eta^3)] + \tfrac{1}{2}[3(1 \pm \varepsilon)\eta - 15(1 \pm \varepsilon)^2\eta^2 + O(\eta^3)] \\ &\quad + \tfrac{1}{3}[\tfrac{3}{2}(1 \pm \varepsilon)^2\eta^2 + O(\eta^3)] + \tfrac{2}{3}[6(1 \pm \varepsilon)^2\eta^2 + O(\eta^3)] \\ &\quad + \tfrac{1}{3}[3(1 \pm \varepsilon)^2\eta^2 + O(\eta^3)] + \tfrac{1}{3}[O(\eta^3)] \\ &= 1 - \tfrac{3}{2}(1 \pm \varepsilon)\eta + \tfrac{5}{2}(1 \pm \varepsilon)^2\eta^2 - O(\eta^3).\end{aligned}$$

Each cluster type with more than 3 triangles contributes $O(\eta^3)$ to the sum and, since Lemma 1 says that we only have to consider the constantly-many cluster types having at most m triangles, all these cluster types together contribute only $O(\eta^3)$ to the sum. Therefore,

$$\Pr[T \text{ is accepted} \mid T \text{ is selected}] = 1 - \tfrac{3}{2}(1 \pm \varepsilon)\eta + \tfrac{5}{2}(1 \pm \varepsilon)^2\eta^2 - O(\eta^3).$$

Actually, at this point we only need to know that this probability is $1 - O(\eta)$, but we will return to this computation when improving the result. It then follows that

$$\Pr[e \in E'] = 1 - \sum_{T \ni e} \Pr[T \text{ is accepted}] = 1 - \tfrac{\eta}{D}(1 - O(\eta)) \deg(e) = 1 - \tfrac{\eta}{D} \deg(e) + O(\eta^2). \quad (2)$$

Now consider the general case, when $F = \{e_1, e_2, \ldots, e_f\}$. We use the Bonferroni inequalities to approximate the complementary probability:

$$\sum_{i=1}^{f} \Pr[e_i \notin E'] - \sum_{i=1}^{f} \sum_{j=1}^{f} \Pr[e_i, e_j \notin E'] \le \Pr\left[\bigcup_{i=1}^{f} (e_i \notin E')\right] \le \sum_{i=1}^{f} \Pr[e_i \notin E'].$$

Looking at the terms of double sum, we see that

$$\Pr[e_i, e_j \notin E'] = \sum_{T_i \ni e_i} \sum_{T_j \ni e_j} \left(\tfrac{\eta}{D}\right)^2 \Pr[T_i \text{ and } T_j \text{ are accepted} \mid T_i \text{ and } T_j \text{ are selected}] \leq (1+\varepsilon)^2\eta^2,$$

since there are at most $(1+\varepsilon)D$ choices for each of T_i and T_j and the probability is trivially at most 1. Thus, the double sum is

$$\sum_{i=1}^{f}\sum_{j=1}^{f} \Pr[e_i, e_j \notin E'] \leq (1+\varepsilon)^2 f^2 \eta^2 = O(\eta^2)$$

and we can conclude that

$$\Pr\left[\bigcup_{i=1}^{f}(e_i \notin E')\right] = \sum_{i=1}^{f} \Pr[e_i \notin E'] - O(\eta^2).$$

To finish up the proof of the lemma, note that

$$\Pr\left[\bigcap_{i=1}^{f}(e_i \in E')\right] = 1 - \Pr\left[\bigcup_{i=1}^{f}(e_i \notin E')\right] = 1 - \sum_{f\in F}\Pr[f \notin E'] + O(\eta^2) = 1 - \tfrac{\eta}{D}\sum_{f \in F}\deg(f) + O(\eta^2).$$

□

Equation (2) also gives us the approxiamte value of H. H was defined as the survival probability of an edge in an ideal graph, where every edge has degree exactly D. Equation (2) proves that

$$H = \Pr[e \in E'] = 1 - \eta + O(\eta^2).$$

For a fixed edge e, we would like to tightly bound the new degree of e. Using Lemma 2, in each case with $|F| = 2$, we see that

$$\begin{aligned}
\mathrm{Ex}[\deg'(e) \mid e \in E'] &= \sum_{T \ni e} \Pr[T \text{ survives} \mid e \in E'] \\
&= \sum_{T=\{e,e_1,e_2\}} (1 - \tfrac{\eta}{D}(\deg(e_1) + \deg(e_2)) + O(\eta^2)) \\
&= (1 - (1 \pm \varepsilon)2\eta + O(\eta^2))\deg(e) \\
&= (1 \pm 2\eta\varepsilon + O(\eta^2))(1 - 2\eta + O(\eta^2))(1 \pm \varepsilon)D \\
&= (1 \pm (\varepsilon + 2\eta\varepsilon + 2\eta\varepsilon^2 + O(\eta^2)))D'.
\end{aligned}$$

Since we assumed further that there was a constant $\gamma > \varepsilon$, we have shown that

$$\mathrm{Ex}[\deg'(e) \mid e \in E'] = (1 \pm (\varepsilon + (1+\gamma)2\eta\varepsilon + O(\eta^2)))D'. \tag{3}$$

We defer the computation of the deviation to the next section. There we will prove the following. Again, the exact constant hidden by the O-notation is independent of s.

Lemma 3 *If $\eta D = \Omega(1)$ then, with probability at least $1 - 3n^{-10}$, $\deg'(e)$ deviates from its expectation by no more than $O\left(\sqrt{\eta D}\log n\right)$.*

Equation (3) and Lemma 3 imply that ε' is at most the solution to the recurrence

$$\varepsilon' = (1 + (1+\gamma)2\eta)\varepsilon + O(\eta^2) + O\left(\sqrt{\eta/D}\log n\right). \tag{4}$$

We aim to solve this recurrence relation, but first we consider the number of surviving edges and the related error factor δ_s. Our goal here is to prove that $\delta_s \leq \varepsilon_s$ ($\delta' \leq \varepsilon'$), but we may have to adjust the constants hidden by the O-notation to make this valid in all cases. We may and do assume inductively that $\delta \leq \varepsilon$.

First the expectation computation for $|E(G')|$: again using Lemma 2, this time with $|F| = 1$, we see that

$$\begin{aligned}
\mathrm{Ex}[|E(G')|] &= \sum_{e \in E(G)} \Pr[e \text{ survives}] \\
&= \sum_{e \in E(G)} (1 - \tfrac{\eta}{D}\deg(e) + O(\eta^2)) \\
&= (1 - (1 \pm \varepsilon)\eta + O(\eta^2))|E(G)| \\
&= (1 \pm \eta\varepsilon + O(\eta^2))(1 - \eta + O(\eta^2))(1 \pm \delta)E \\
&= (1 \pm (\delta + \eta\varepsilon + \eta\varepsilon\delta + O(\eta^2)))E'.
\end{aligned}$$

Since we assumed further that there was a constant $\gamma > \delta$, we have shown that

$$\mathrm{Ex}[|E(G')|] = (1 \pm (\delta + (1+\gamma)\eta\varepsilon + O(\eta^2)))E'. \tag{5}$$

Again we defer the computation of the deviation to the next section.

Lemma 4 *If $\eta E = \Omega(1)$ then, with probability at least $1 - 3n^{-10}$, $|E(G')|$ deviates from its expectation by no more than $O\left(\sqrt{\eta E}\log n\right)$.*

Equation (5), Lemma 4, and our inductive assumption that $\delta \leq \varepsilon$ imply that δ' is at most

$$\begin{aligned}
\delta' &= \delta + (1+\gamma)\eta\varepsilon + O(\eta^2) + O\left(\sqrt{\eta/E}\log n\right) \\
&\leq \varepsilon + (1+\gamma)2\eta\varepsilon + O(\eta^2) + O\left(\sqrt{\eta/D}\log n\right).
\end{aligned}$$

We would like to say that this last expression is at most the ε' of (4), but this might not be true due to the constants hidden in the O-notation. But this is no problem: simply redefine ε' using whichever constants are greater, those given here or those in (4). With this new ε', we may safely conclude that $\delta' \leq \varepsilon'$.

Returning to a broader perspective, (4) (possibly with modified hidden constants) says that ε_s is less than the solution to the recurrence

$$\varepsilon_s = \varepsilon_{s-1} H^{-(1+\gamma)2} + O(\eta^2) + O\left(\sqrt{\frac{\eta}{n}} H^{-(1+\gamma)s} \log n\right), \tag{6}$$

where, as previously noted, $H = 1 - \eta + O(\eta^2)$.

Recurrences, such as this one, of the form

$$x_s = A^2 x_{s-1} + B + CA^s$$

have the solution

$$x_s = B\frac{A^{2s}-1}{A^2-1} + CA^s\frac{A^s-1}{A-1}.$$

For us, this means that

$$\begin{aligned}\varepsilon_s &= O\left(\eta^2 \frac{H^{-(1+\gamma)2s}}{(1+\gamma)2\eta}\right) + O\left(\sqrt{\frac{\eta}{n}} H^{-(1+\gamma)s} \frac{H^{-(1+\gamma)s}}{(1+\gamma)\eta} \log n\right) \\ &= O\left(\eta H^{-(1+\gamma)2s}\right) + O\left(\frac{\log n}{\sqrt{\eta n}} H^{-(1+\gamma)2s}\right).\end{aligned}$$

We were very careful to point out that the hidden constants are independent of s in order to insure that B and C are also. This guarantees that this computation is valid.

We would like to take as many steps as possible, always ensuring that $\varepsilon_s \leq \gamma$. Since $\{\varepsilon_s\}$ forms an increasing sequence, it is only necessary that

$$\varepsilon_t = O\left(\eta H^{-(1+\gamma)2t}\right) + O\left(\frac{\log n}{\sqrt{\eta n}} H^{-(1+\gamma)2t}\right) \leq \gamma. \tag{7}$$

This gives us enough information to set η and t intelligently. In order that both terms of (7) be roughly equal, set

$$\eta = n^{-1/3+\gamma}.$$

Condition (7) is then implied by

$$H^{-t} \leq n^{1/6-\gamma}$$

for sufficiently large n. This inequality is easily satisfied with equality by setting

$$t = \frac{(1/6-\gamma)\log n}{\log(1/H)} = \Theta\left(\frac{\log n}{\eta}\right) \ll n^{1/3}.$$

This verifies the first part of assertion (1). The second part is valid since

$$\eta D_t = \eta H^{2t} n = n^{1/3+3\gamma} \gg 1.$$

Turning finally to the number of edges remaining at this step t, we see that

$$|E_t| = O\left(H^t n^2\right) = O\left(n^{11/6+\gamma}\right).$$

We can force the number of remaining edges as close to $n^{11/6}$ as we like by fixing $\gamma > 0$ sufficiently small. In other words, modulo Lemmas 3 and 4, we've proven

Theorem 5 *For any fixed $\gamma > 0$, the random greedy triangle packing algorithm almost always leaves fewer than $O\left(n^{11/6+\gamma}\right)$ uncovered pairs.*

4 Deviations

To bound the deviations of the random variables discussed in the previous section, we'll need the powerful large deviation inequality discussed in full detail in [8]. Here we only give the essential definitions and relevant considerations.

Assume we have a probability space generated by independent random variables X_i (choices), where choice X_i is from the finite set A_i, and a function (random variable) $Y = f(X_1, \ldots, X_n)$ on that probability space. We are interested in proving a sharp concentration result on Y—that is, to bound $\Pr[|Y - \mathrm{Ex}[Y]| > a]$, for any a, as well as we can.

Consider the following *query game*, the aim of which is to determine the value of Y. We can make queries of the form "what is the value of X_i?" in any order we want. The questioning can be adaptive, meaning that we can chose the next X_i to be queried as a function of the knowledge gained so far. A *querying strategy* for Y is an algorithm for determining which queries should be made in which order. We can describe a strategy as a decision tree whose internal nodes designate queries to be made. If a given node represents the query "what is the value of X_i?" then it has as many children as there are possible values for X_i. You might think of the down edges as being labelled with the values $a \in A_i$. In this fashion, every path from the root to a node going through vertices corresponding to $X_{i_1}, \ldots, X_{i_k}$ defines an assignment $a_1, \ldots, a_k$ for these random variables. We can then think of the j^{th} node as storing the value $\mathrm{Ex}[Y \mid X_{i_1} = a_1, \ldots, X_{i_{j-1}} = a_{j-1}]$. In particular, the leaves of the tree store each of the possible values of Y.

The idea behind this query-based large deviation inequality is to bound the deviation of Y by carefully bounding the contribution to the deviation made by each X_i. A strategy efficiently expresses which X_i are relevant and more importantly allows us to estimate the contribution at the time that the query is made. For instance, the result of previous queries may imply that the next query will have a more limited effect than would have been initially anticipated. Formally, define the *variance of a query* q concerning choice i to be

$$v_q = \sum_{a \in A_i} \Pr[X_i = a]\, \mu_{q,a}^2,$$

where

$$\mu_{q,a} = \mathrm{Ex}[Y \mid X_i = a \text{ and previous queries}] - \mathrm{Ex}[Y \mid \text{previous queries}].$$

In words, $\mu_{q,a}$ measures the amount which our expectation changes when the answer to query q is revealed to be a.

Also define the *maximum effect* of query q as

$$c_q = \max_{a,b \in A_i} |\mu_{q,a} - \mu_{q,b}|.$$

A way to think about c_q is the following. If we consider the children of node q, c_q is the maximum difference between any two values $\mathrm{Ex}[Y \mid \text{previous queries}]$ stored at the children. In the sequel, we will always compute an upper bound on c_q by taking the maximum amount which Y can change if choice i is changed, but all other choices remain the same. That is, we consider the subtree rooted at q and consider the maximum difference between any two values stored in the leaves of this subtree.

When computing v_q, we use two elementary but non-trivial upper bounds. For queries with two possible outcomes YES and NO (or two types of equivalent outcomes),

$$v_q \le \Pr[\text{YES}] \Pr[\text{NO}]\, c_q^2.$$

In general,

$$v_q \le c_q^2/4.$$

Turing back to the whole strategy, the *variance of a strategy* is the maximum cumulative variance of any sequence of queries which determines Y. In other words, we sum the variances down every path leading from the root to the leaves of the decision tree and take the maximum sum. By the way, the use of the term variance is meant to be suggestive: the variance of a strategy for determining Y is an upper bound on the variance of Y.

There is only one more twist to the story: sometimes it can happen that very unlikely situations occur which force certain queries to have very large variances driving up the variance of the entire strategy. We call such rare but ruinous sets of values for the X_i *exceptional outcomes.* The Theorem is able to give tight bounds on the deviation of Y despite the presence of exceptional outcomes, as long as their probability is not too great.

Theorem 6 *Let m and M be, respectively, the minimum and maximum values taken by Y over all possible outcomes. Let $\mathcal{C}$ be a set of exceptional outcomes. Consider a strategy for determining Y, assuming that the actual outcome is not in $\mathcal{C}$. If the variance of this strategy is at most V then*

$$\Pr\left[|Y - \mathrm{Ex}[Y]| > 2\sqrt{\varphi V} + (M-m)\Pr[\mathcal{C}]\right] \le 2e^{-\varphi} + \Pr[\mathcal{C}],$$

for every $0 \le \varphi \le V/\max c_q^2$.

Now we can return to the matter at hand, proving Lemmas 3 and 4. To prove Lemma 3, start by fixing an edge e and assuming that it survives. That just means that none of its incident triangles is accepted. But some of them may not survive as triangles in G', since some of their edges may be covered by accepted triangles.

The relevant probability space is formed from the independent triangle selections and independent, uniformly random orderings for each cluster. At first glance, the fact that the orderings depend on which triangles were selected, and hence which clusters exist to be ordered, seems to preclude the use of Theorem 6. But instead of picking an ordering for each cluster, we simply pick an ordering of all triangles uniformly at random. It is easy to see that a random ordering of all triangles, when restricted to the triangles of a single cluster, is a uniform random ordering and that the orderings of necessarily disjoint clusters are independent.

Each of the $2\deg(e)$ edges in triangles incident to e can only disappear if it is covered by a cluster, so we should first fully explore all clusters covering these edges. Note trivially that each edge is covered by at most one cluster.

To explore the cluster at one edge takes at most $(2m+1)(1+\varepsilon)D = O(D)$ queries, where $m = O(1)$ is the maximum size of a cluster (Lemma 1). This gives a total of at most $O(D^2)$ queries. These are YES/NO queries, with $\Pr[\text{YES}] = \eta/D$. Importantly, the effect of a query is at most the number of edges covered by the cluster. This is certainly less than $2m+1 = O(1)$. Altogether, these queries contribute at most $O(D^2)\frac{\eta}{D}O(1)^2 = O(\eta D)$ to the total variance.

Next we have to query the ordering. This is equivalent to querying the independent orderings for each cluster, now that we know what they are. Each cluster ordering can affect only the edges covered by that cluster, so the effect of each query is $O(1)$.

The only question is, how many ordering queries do we have to make? Well, clearly the answer is the number of clusters covering edges in triangles incident with e. This could conceivably be as much as $O(D)$, which would lead to a weaker deviation result than we would like, so we use the "exceptional outcomes" feature of Theorem 6.

Edge f is covered by a cluster with probability $1-(1-\eta/D)^{\deg(f)} = O(\eta)$, so the expected number of relevant edges covered by clusters is $O(\eta D)$. Furthermore, if we ignore the triangles incident with e, the events "f is covered by a cluster", for the relevant f's, are independent, so the Chernoff bounds imply that the number of such edges covered by clusters is not much more than its expectation. In particular, this number is with probability at least $1-n^{-10}$ no more than $O(\eta D\log n)$. And of course, the number of clusters is no more than the number of edges covered by clusters. If we then consider the triangles incident with e, we note that these triangles can increase the number of clusters by at most 1 and may, in fact, reduce it by joining previously isolated clusters.

Thus, we say that an outcome is exceptional if it results in more than $O(\eta D\log n)$ clusters covering edges in triangles incident with e. So now we know that in non-exceptional circumstances we need never make more than $O(\eta D \log n)$ cluster ordering queries. This gives as total variance for this strategy

$$V = O(\eta D) + O(\eta D \log n) \cdot O(1)^2/4 = O(\eta D \log n).$$

We want the probability of deviation to be on the order of n^{-10}, so we put $\varphi = 10 \log n$ and get a deviation of $O\left(\sqrt{\eta D} \log n\right)$, as claimed in Lemma 3. Note that the side condition of Theorem 6, that $0 \leq \varphi \leq V/\max c_q^2$, only requires that $\eta D = \Omega(1)$.

Finally we prove Lemma 4. Here we're concerned with the number of surviving edges in the entire graph, so we have to first query all $O(ED)$ triangles. As before, these are YES/NO queries with $\Pr[\text{YES}] = \eta/D$ and maximum effect $2m + 1 = O(1)$. These queries therefore contribute $O(ED)\frac{\eta}{D}O(1)^2 = O(\eta E)$ to the total variance.

Querying the ordering is equivalent to querying the independent orderings of all clusters in the graph. With very small probability this is quite large, but an argument like the one given above shows that with probability at least $1 - n^{-10}$ there are no more than $O(\eta E \log n)$ clusters. Each cluster ordering affects only the $O(1)$ edges in that cluster, so in non-exceptional circumstances the cluster ordering queries contribute $O(\eta E \log n) \cdot O(1)^2/4 = O(\eta E \log n)$ to the total variance.

The strategy therefore has total variance $V = O(\eta E \log n)$. Setting $\varphi = 10 \log n$ to get a deviation probability of $3n^{-10}$, the large deviation inequality gives a deviation of order $O\left(\sqrt{\eta E} \log n\right)$, as claimed in Lemma 4. The side condition requires that $\eta E = \Omega(1)$.

5 Improvements

Now we would like to sharpen the proof of Theorem 5 as much as possible to get a better bound. Basically this amounts to doing all the computations as accurately as possible. We won't actually do the computations, but rather show how one would do them and look at their consequences.

The first thing to do would be to improve Lemma 2 by detailing the $O(\eta^2)$ term. That is done by using an extended version of Table 1 to work out a more accurate expression for $\Pr[T \text{ is accepted} \mid T \text{ is selected}]$. As we've already shown in the proof of Lemma 2, the first three terms are, for instance,

$$\Pr[T \text{ is accepted} \mid T \text{ is selected}] = 1 - \tfrac{3}{2}(1 \pm \varepsilon)\eta + \tfrac{5}{2}(1 \pm \varepsilon)^2\eta^2 - O(\eta^3).$$

We can continue this to any desired accuracy $O(\eta^b)$ as long as we know how many potential clusters of each type there are. It is for this reason that Lemma 2 is set up to handle any constant number of triangles (we know that we never need to consider clusters with more than a constant number of triangles by Lemma 1).

Continuing, it follows that we can work out $\Pr[e \in E']$ to any accuracy $O(\eta^b)$. For instance,

$$\begin{aligned}\Pr[e \in E'] &= 1 - \sum_{T \ni e} \Pr[T \text{ is accepted}] \\ &= 1 - \eta \deg(e)/D + \tfrac{3}{2}(1 \pm \varepsilon)\eta^2 \deg(e)/D - \tfrac{5}{2}(1 \pm \varepsilon)^2 \eta^3 \deg(e)/D + O(\eta^4).\end{aligned}$$

In the ideal case where all edges have degree exactly D, this probability is

$$H := \Pr[e \in E'] = 1 - \eta + \tfrac{3}{2}\eta^2 - \tfrac{5}{2}\eta^3 + O(\eta^4),$$

so for a real edge e we can conclude that for any $b \geq 1$,

$$\Pr[e \in E'] = H \pm \varepsilon\eta(1 + 4\eta) + O(\eta^b)$$

For any three distinct edges e_1, e_2, and e one can show that

$$\begin{aligned}\Pr[e_1, e_2 \in E' \mid e \in E'] &= (H \pm \varepsilon\eta(1 + 4\eta))^2 + O(\eta^b) \\ &= H^2 \pm 2\varepsilon\eta(1 + 5\eta) + O(\eta^b).\end{aligned}$$

Thus,

$$\begin{aligned}\mathrm{Ex}[\deg'(e) \mid e \in E'] &= \sum_{T \ni e} \Pr[T \text{ survives} \mid e \in E'] = \sum_{T = \{e, e_1, e_2\}} \Pr[e_1, e_2 \in E' \mid e \in E'] \\ &= (1 \pm \varepsilon)\left(H^2 \pm 2\varepsilon\eta(1 + 5\eta) + O(\eta^b)\right) D \\ &= (1 \pm \varepsilon)\left(1 \pm 2\varepsilon\eta(1 + 5\eta) + O(\eta^b)\right) H^2 D \\ &= \left(1 \pm \varepsilon + 2\varepsilon(1 + \varepsilon)\eta(1 + 5\eta) + O(\eta^b)\right) D'.\end{aligned}$$

This and Lemma 3 give

$$\begin{aligned}\varepsilon' &= (1 + (1 + \varepsilon)2\eta(1 + 5\eta))\varepsilon + O(\eta^b) + O\left(\sqrt{\eta/D}\log n\right) \\ &\leq \varepsilon H^{-(1+\gamma)2} + O(\eta^b) + O\left(\sqrt{\eta/D}\log n\right).\end{aligned}$$

This says that ε_s is less than the solution to the recurrence

$$\varepsilon_s = \varepsilon_{s-1} H^{-(1+\gamma)2} + O(\eta^b) + O\left(\sqrt{\frac{\eta}{n}} H^{-(1+\gamma)s} \log n\right).$$

This is identical to recurrence equation (6) except that $O(\eta^2)$ is replaced by $O(\eta^b)$. Again, the constants hidden by the O-notation are independent of s, so we may solve this new recurrence relation and get

$$\varepsilon_s = O\left(\eta^{b-1} H^{-(1+\gamma)2s}\right) + O\left(\frac{\log n}{\sqrt{\eta n}} H^{-(1+\gamma)2s}\right).$$

All we have to do is set η and t so that

$$\varepsilon_t = O\left(\eta^{b-1} H^{-(1+\gamma)2t}\right) + O\left(\frac{\log n}{\sqrt{\eta n}} H^{-(1+\gamma)2t}\right) \leq \gamma. \tag{8}$$

This is accomplished first by setting

$$\eta = n^{-(1-\gamma)/(2b-1)}.$$

Condition (8) is then implied by

$$H^{-t} \leq n^{(1/4)(1-2\gamma)(1-1/b)},$$

for sufficiently large n. This is satisfied with equality by setting

$$t = \frac{(1/4)(1-2\gamma)(1-1/b)\log n}{\log(1/H)} = \Theta\left(\frac{\log n}{\eta}\right) \ll n^{1/(2b-1)} \leq n^{1/3}.$$

This verifies the first part of assertion (1). The second part is again valid since

$$\eta D_t = \eta H^{2t} n = n^{1-(1-\gamma)/(2b-1)-(1/2)(1-2\gamma)(1-1/b)} \gg n^{1/6} \gg 1.$$

Finally, the number of edges remaining after the t^{th} burst is

$$|E_t| = O\left(H^t n^2\right) = O\left(n^{2-(1/4)(1-2\gamma)(1-1/b)}\right).$$

We can force the exponent to be as close to 7/4 as we like by setting γ sufficiently small and b sufficiently large. Therefore, we've proven the following theorem.

Theorem 7 *For any $\xi > 0$, the random greedy triangle packing algorithm almost always leaves fewer than $O\left(n^{7/4+\xi}\right)$ uncovered pairs.*

6 Packings in Hypergraphs

Actually, everything done in this paper can be done in the more general setting of packings in simple, k-uniform, D-regular hypergraphs on N points. A hypergraph is *simple*, *k-uniform*, and *D-regular* if no two edges share two points, if all edges contain exactly k points, and if every point is contained in D edges. A *packing* is a collection of disjoint edges.

The random greedy hypergraph packing algorithm picks edges one at a time, uniformly at random to build up a maximal packing. How many points are not covered by any edge in the resulting packing? The results of Spencer [13] and Rödl and Thoma [12] apply to this setting and state that almost always the number of uncovered points is $o(N)$. The corresponding conjecture is that the number of uncovered points is almost always $ND^{-1/(k-1)+o(1)}$. The techniques of this paper can be used to show that the number of uncovered points is almost always $ND^{-1/2(k-1)+o(1)}$.

To see how triangle packing is a special case of hypergraph packing, we construct a hypergraph whose packings correspond to partial triple systems. The $N = \binom{n}{2}$ points of the hypergraph are the pairs of points in the triple system and the $\binom{n}{3}$ hyperedges correspond to all of the possible triples. A packing is then a collection of triples no two of which share a pair—that is, a partial triple system. We note that a similar construction gives a hypergraph whose packings correspond to partial designs with larger parameters.

Now it is clear why we called the number of triangles containing a given edge that edge's degree: in the hypergraph the edge is a point and the triangles which contain it are its incident hyperedges. Also, in this setting the clusters are simply the connected components of the hypergraph induced by the selected hyperedges.

Acknowledgements

Special thanks are due to Joel Spencer for suggesting the problem and to him and the anonymous referee for useful comments on an earlier version of the paper.

References

[1] N. Alon, H.J. Kim, J. Spencer, Nearly perfect matchings in regular simple hypergraphs, *Israel Journal of Mathematics*, to appear.

[2] K.T. Balińska and I. Wieczorek, Random triangle deletion process, Technical University Poznań, 1989.

[3] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Cambridge University Press, Cambridge, 1986.

[4] P. Erdős and H. Hanani, On a limit theorem in combinatorial analysis, *Publ. Math. Debrecen* **10** (1963), 10–13.

[5] P. Frankl and V. Rödl, Near perfect coverings in graphs and hypergraphs, *European Journal of Combinatorics* **6** (1985), 317–326.

[6] D. Gordon, G. Kuperberg, O. Patashnik, and J. Spencer, Asymptotically optimal covering designs, *Journal of Combinatorial Theory, Series A*, to appear.

[7] D.A. Grable, More-than-nearly-perfect packings and partial designs, preprint, 1994.

[8] D.A. Grable, A large deviation inequality for functions of independent, multi-way choices, *Combinatorics, Probability, and Computing*, to appear.

[9] J. Kahn, Asymptotically good list-colorings, *Journal of Combinatorial Theory, Series A* **73** (1996), 1–59.

[10] N. Pippenger and J. Spencer, Asymptotic behavior of the chromatic index for hypergraphs, *Journal of Combinatorial Theory, Series A* **51** (1989), 24–42.

[11] V. Rödl, On a packing and covering problem, *European Journal of Combinatorics* **5** (1985), 69–78.

[12] V. Rödl and L. Thoma, Asymptotic packing and the random greedy algorithm, *Random Structures and Algorithms* **8** (1996), 161–177.

[13] J. Spencer, Asymptotic packing via a branching process, *Random Structures and Algorithms* **7** (1995), 167–172.