Random graph processes
and optimisation

Julie Anne Cain

Submitted in total fulfillment of the
requirements of the degree of
Doctor of Philosophy

January, 2006

Department of Mathematics and Statistics,
University of Melbourne
Abstract

Random graph processes are most often used to investigate theoretical questions about random graphs. A randomised algorithm can be defined specifically for the purpose of finding some structure in a graph, such as a matching, a colouring or a particular kind of subgraph. Properties of the related random graph process then suggest properties, or bounds on properties, of the structure. In this thesis, we use a random graph process to analyse a particular load balancing algorithm from theoretical computer science. By doing so, we demonstrate that random graph processes may also be used to analyse other algorithms and systems of a random nature, from areas such as computer science, telecommunications and other areas of engineering and mathematics. Moreover, this approach can lead to theoretical results on the performance of algorithms that are difficult to obtain by other methods. In the course of our analysis we are also led to some results of the first kind, relating to the structure of the random graph.

The particular algorithm that we analyse is a randomised algorithm for an off-line load balancing problem with two choices. The load balancing algorithm, in an initial stage, mirrors an algorithm which finds the $k$-core of a graph. This latter algorithm and the related random graph process have been previously analysed by Pittel, Spencer and Wormald [31], using a differential equation method, to determine the thresholds for the existence of a $k$-core in a random graph. We modify their approach by using a random pseudograph model due to Bollobás and Frieze [7], and Chvátal [10], in place of the uniform random graph. This makes the analysis somewhat simpler, and leads to a shortened derivation of the thresholds and other properties of $k$-cores.
An extension of this analysis leads to probabilistic results on the performance of the load balancing algorithm. By employing again the differential equation method, in conjunction with some branching process and graph theoretic arguments, we show that the load balancing algorithm is asymptotically almost surely optimal.

In doing so, we also provide a rigorous proof of the threshold for $k$-orientability of the random graph, investigated by Karp and Saks [23].

The final chapter of the thesis is concerned with the development and analysis of heuristic algorithms for a different optimisation problem; the $d$-dimensional Euclidean Steiner tree problem. This represents some early work undertaken by the candidate, and the approach to analysing the algorithms is quite different to that in the first part of the thesis.

We present a family of new, polynomial time heuristics based on minimum spanning tree (MST) algorithms, but incorporating a geometric optimisation step to achieve vastly improved performance. We lower bound the worst case performance of these algorithms by a simple, direct proof, and demonstrate by simulation that the average case performance is good in comparison to the MST and to a slow running metropolis algorithm that is conjectured to have a near optimal output.

In conjunction with the first part of the thesis, this work highlights that although an algorithm may perform well statistically, rigorous, tight bounds on the performance of heuristic algorithms for NP-hard optimisation problems, can be difficult to achieve. A creative approach, using a variety of techniques can sometimes yield interesting results.
Declarations

This is to certify that:

(i) the thesis comprises only my original work towards the PhD except where indicated in the Preface,

(ii) due acknowledgement has been made in the text to all other material used,

(iii) the thesis is less than 100,000 words in length, exclusive of tables, bibliographies and appendices.

Julie Cain
Preface

This thesis is submitted to the University of Melbourne in support of my application for admission to the degree of Doctor of Philosophy. No part of it has been submitted in support of an application for another degree or qualification of this or any other institution on learning.

Parts of the thesis appear in the following papers. The work on the topic of the $k$-core of the random graph, in particular that which culminates in Theorem 5.1, is joint work with Nicholas Wormald and appears in the paper “Encores on cores”.

Acknowledgments

First and foremost I would like to thank my supervisor Professor Nick Wormald for inspiring me to learn new things and for continuing to provide supervision and support from afar after taking up a new post in Canada. I would also like to thank Nick and his wife Hania for their generous hospitality during my stays in Waterloo. Thanks also to Professor Hyam Rubinstein, Dr. Catherine Greenhill and Dr. Sanming Zhou who, at various times, stepped into the breach in Melbourne to provide advice and encouragement.

Thank you to my family for being there and not asking too many questions, and thank you to the friends and office mates who could always be found in dire moments. You are many, and I hope to thank you each in person. Finally, thank you to Andrew because it all started with you.

“\textit{The only real voyage of discovery consists not in seeing new landscapes, but in having new eyes, in seeing the universe with the eyes of another, of hundreds of others, in seeing the hundreds of universes that each of them sees.}” – Marcel Proust.
Contents

Abstract i

Declarations iii

Preface iv

Acknowledgments v

0 Introduction 1

1 Preliminaries 7

1.1 Load balancing 7

1.2 The $k$-core 9

1.3 The graph model 10

1.4 Some notation and definitions 14

1.4.1 Random graphs and processes 14

1.4.2 Probability 16

1.4.3 Asymptotics 17

1.4.4 The differential equation method 17
1.4.5 Branching processes .................................................. 20
1.5 Organisation of thesis .................................................. 22

2 The algorithms for pseudographs ................................. 24
2.1 Algorithm for the $k$-core ............................................. 25
2.2 Algorithm for load balancing ......................................... 26

3 The conditioned pseudograph model ......................... 29
3.1 The allocation model ................................................... 29
3.2 The conditioned allocation model ................................. 32
3.3 Simple graphs .......................................................... 45

4 The pseudograph processes ........................................... 47
4.1 Edge deletion and orientation steps ................................. 49
4.2 Random pairings ........................................................ 51
4.3 Heavy vertices .......................................................... 52
4.4 Transition probabilities ................................................. 56
4.4.1 Transition probabilities for the $k$-core process .............. 57
4.4.2 Transition probabilities for the load balancing process .... 63
4.5 Partial pre-allocations .................................................. 68
4.6 The partial pre-allocation processes ............................... 76
4.6.1 The $k$-core process ................................................ 76
4.6.2 The load balancing process ...................................... 80

5 Analysis of the $k$-core process ................................ 83
Introduction

This thesis is concerned with certain optimisation problems, some heuristic algorithms used to solve them, and methods for analysing the performance of these algorithms. The first part focuses on the use of random graph processes to achieve a theoretical, average-case analysis of the performance of algorithms for these problems. We analyse an algorithm for an off-line load balancing problem, and, in the course of this analysis achieve some more general results on random graphs and other random structures and processes. In the last chapter of the thesis we present some heuristic algorithms for another optimisation problem, and take quite a different approach to analysing them. We first take a worst-case approach, and find a theoretical lower bound for the performance of the algorithms, and then perform an average-case analysis experimentally, via simulation.

In [35], Sanders proposes a linear time algorithm for allocating file requests in a large system of discs, where there are two copies of each file uniformly distributed on the system. Such a system can be modelled as a graph, and Sanders’ algorithm translates into an algorithm that generates an orientation for each edge, attempting to minimise the maximum in-degree of vertices in the graph. The allocation of an edge to a vertex is represented by orienting the edge towards the vertex to which it is allocated (one of its end vertices). For fixed integer \( k \) (the maximum load), we consider an algorithm which attempts to orient the edges of a graph so that no vertex has in-degree greater than \( k \).

The research presented here was motivated by the relationship between this load balancing algorithm and the \( k \)-core of a graph: Sanders’ algorithm, for load \( k \), prioritises vertices of
degree \( k \) or less for allocations. By doing this it emulates, in an initial ‘phase’, an algorithm which finds the \((k + 1)\)-core of the graph. This \( k \)-core algorithm, via a corresponding random graph process, was analysed by Pittel, Spencer and Wormald [31], resulting in, for the first time, exact thresholds for the appearance of a \( k \)-core in a random graph, as well as asymptotic formulae for the size of \( k \)-cores. It was anticipated that a similar approach applied to the load balancing algorithm may lead to results on the performance of that algorithm. This it did, and more along the way.

Preliminary material on load balancing, \( k \)-cores, random graphs and random graph processes is contained in Chapter 1. Also included is a more detailed description of some features of the problem, and the relevant notation.

The \( k \)-core and load balancing algorithms are analysed by considering the two discrete, random processes determined by these algorithms applied to a random pseudograph model, due to Bollobás and Frieze [7], and Chvátal [10]. Each process is indexed by \( n \), the size of the pseudograph on which it is defined. In the limit as \( n \) goes to infinity, certain random variables for the process exhibit concentrated behaviour, which implies that parameters of the output of the algorithm tend to certain values, with high probability. In fact, we can find thresholds for the performance of an algorithm. This is a value for some parameter, \( r \), of the input such that, if \( r \) is significantly less than the threshold, then the probability of some event for the output of the algorithm tends to one, and if \( r \) is significantly greater than the threshold, the probability tends to zero (or visa versa) as \( n \) goes to infinity. We say that an event holds asymptotically almost surely (a.a.s.) if the probability of that event tends to one as \( n \) goes to infinity.

This pseudograph model is considerably easier to work with than the uniform random graph, for a number of reasons, but mainly because each edge, \((i, j)\), in the pseudograph is uniformly distributed on \([n] \times [n]\), independently of every other edge. This simplifies enumeration formulae, the distributions of vertex degrees, and the transition probabilities for edge deletion and similar processes, all of which we exploit here. Moreover, in [10], Chvátal showed that properties which hold a.a.s. for random pseudographs also hold a.a.s.
for the uniform random graph. Thus analysing the processes for random pseudographs also yields the results for the random graph.

The \(k\)-core and load balancing algorithms are described in Chapter 2, and in Chapter 3, the pseudograph model and some important properties are given. We also define a restricted space of pseudographs for which the degree sequence of low degree vertices is fixed. We show that the degree sequence of high degree vertices in this model has a truncated multinomial distribution, and that the distributions of vertex degrees are exactly independent, truncated Poissons, conditioned on their sum being twice the number of edges. Hence, we show that the degrees of high degree vertices are individually, asymptotically truncated Poisson.

In Chapter 4 we describe in detail some features of the processes. In particular, we show that, subject to the deletion of a randomly chosen edge adjacent to a low degree vertex, the asymptotically truncated Poisson distribution of vertex degrees (for high degree vertices) is maintained. This has a number of consequences. For purposes here, it means that the processes can be analysed as Markov chains in which the states are the restricted spaces of pseudographs, and the transition probabilities between states are determined by a simple set of parameters of the restricted spaces. More generally, it implies that the distribution of vertex degrees in the \(k\)-core of the random pseudograph is exactly truncated multinomial, and that, conditioning on the number of vertices and edges, the \(k\)-core is distributed precisely as the restriction of the pseudograph model to pseudographs with minimum degree \(k\). Such properties of \(k\)-cores can be quite useful as cores sometimes arise in the analysis of other algorithms and processes. This result is used, for example, in [5].

In Chapter 5 we prove our main theorem on cores, Theorem 5.1, which gives the threshold for the existence of a \(k\)-core in a random graph, originally found in [31], and more recently by other authors using different approaches [21, 26, 30]. The distribution of the degree sequence of the \(k\)-core also arises naturally from our arguments, as do asymptotic formulae for the size and number of edges in the \(k\)-core, and for determining the threshold for a
random graph to have a $k$-core with a certain average degree. The latter is needed for our result on the load balancing problem. The result about edges can also be found in [31], although requires some searching through the proof.

Our proof is somewhat shorter and simpler than in [31], although sharper error terms are obtained in that paper. We take a similar approach, which involves finding continuous approximations to random variables of the process, by solving a system of differential equations suggested by the expected changes of the random variables in each “step” of the process. Then we apply general purpose theorems for the use of this method, developed by Wormald [40, 41]. The main simplification in our re-derivation is achieved by using random pseudographs rather than graphs. By doing so, we avoid the use of some asymptotic enumeration formulae for simple graphs, and having to analyse other error terms that arise as a consequence of the edges in a simple graph not being completely independent. Interestingly, this pseudograph model is used in [31] to estimate the distributions of some properties of the degree sequence of the random graph, but not to the full extent it is here.

If a graph or pseudograph has no $(k + 1)$-core, then Sanders’ algorithm will successfully find an orientation with maximum in-degree no greater than $k$ [12]. Hence, the threshold for the existence of a $(k + 1)$-core is a lower bound for the threshold for Sanders’ algorithm to succeed with max load $k$. On the other hand, define a graph, $G$, to be $k$-orientable if there exists an orientation of the edges of $G$ in which no vertex has in-degree greater than $k$. It follows from Hall’s Theorem that (Lemma 1.1) a graph or pseudograph, $G$, is $k$-orientable if and only if $G$ contains no subgraph with average degree greater than $2k$. Hence, the thresholds for the existence of, firstly, a subgraph of average degree $2k$, and secondly, a $(k + 1)$-core with average degree $2k$, are upper bounds for the threshold for Sanders’ algorithm to succeed.

Karp and Saks have investigated the question of whether the threshold for $k$-orientability of the random graph, which by Lemma 1.1 is the threshold for the graph to have a subgraph of average degree greater than $2k$, is the same as the threshold for the $(k + 1)$-core to have average degree $2k$. As yet, they have no rigorous proof of the conjecture [24]. In Chapters 6
and 7 we show that a.a.s., conditional on the \((k + 1)\)-core having average degree less than \(2k\), Sanders’ algorithm will succeed (Lemma 6.3). By the reasoning above, this implies our main result, Theorem 6.1 and its corollary, that Sanders’ algorithm is asymptotically almost surely optimal. As a further corollary, this also provides a rigorous proof of Karp and Saks conjecture (Corollary 6.2).

The crux of the proof of Lemma 6.3 relies on the analysis of two particular random variables. We show that for Sanders’ algorithm to succeed it is necessary for the expected difference between these two variables to remain positive (bounded away from zero) throughout the process. In Chapter 6, the differential equation method is used again to approximate the behaviour of these variables, however it does not give the entire argument. Some variables, on which our two main variables depend, have near zero expectation, so we bound the probability of events related to the change in value of these variables by using branching process arguments. We also exploit the truncated Poisson distribution of vertex degrees to find a.a.s. bounds on the values of certain other variables, conditioned on certain events. This combination of arguments seems to point to the claim of Lemma 6.3, however, to analyse the ‘end’ of the process, when all the variables become small in expectation, quite different arguments are required.

In Chapter 7, the end of the process is analysed by considering more directly what happens in a pseudograph subjected to Sanders’ algorithm. What we show is roughly equivalent to showing that, a.a.s., conditioned on a certain event a pseudograph contains no ‘bad’ subgraph, that is, one that does not get successfully processed by algorithm (for example, a subgraph of average degree \(2k\)). We define an auxiliary algorithm by which Sanders’ algorithm generates a colouring and matching of points in a pseudograph, as it proceeds. This matching will identify any part of the pseudograph containing a bad subgraph, if it exists. Then we analyse the discrete, random process determined by this auxiliary algorithm and, by considering an associated branching process, show that a.a.s. no such bad subgraph exists.

In Chapter 8 we discuss some open problems related to Sanders’ algorithms and some
possible directions for further research.

In the final chapter, Chapter 9, we present some heuristic algorithms for the $d$-dimensional Euclidean Steiner tree (EST) problem, and we analyse the performance of these algorithms via a quite different approach to that in the first part of the thesis. We focus on the 3 and higher dimensional problem, which has received a lot less attention than the $d = 2$ case and is considerably more difficult. The $d$-dimensional EST problem is to find the network with minimum Euclidean length spanning a fixed set of terminals in $d$-space, allowing the addition of auxilliary (Steiner) points to the set. The problem is well understood in the plane but has been largely ignored in higher dimensions where there are various applications, and shortcuts useful in the $d = 2$ case cannot be applied.

The EST problem is known to be NP-hard, giving rise to the search for and study of effective, polynomial time heuristics. Smith and Shor’s simple “greedy tree”, based on the Minimum Spanning Tree (MST), is considered and by incorporating a geometric optimisation step, a family of related polynomial time heuristics is presented. We first take a worst-case approach and prove that these heuristics all produce spanning networks no longer than the MST. Then we show, using simulation, that for $d = 3$, the average performance is favorable in comparison to the MST and to a slow running metropolis algorithm that is conjectured to have a near optimal output.

An index of symbols and notation and an index of terminology are included as appendices.
Chapter 1

Preliminaries

In this chapter we present background material on load balancing, $k$-cores, random graphs and random graph processes. We highlight some features of our approach to the problem, and include the relevant notation. Preliminary material and definitions for the Euclidean Steiner tree problem are deferred until the start of Chapter 9.

1.1 Load balancing

Load balancing refers to a general class of problems in which a method for assigning a large number of requests or tasks to a set of processors or sites is sought, with the aim of minimizing the load on any single site. The particular load balancing problem of interest here is concerned with requests for data, or files, contained in a large system of discs. One or more copies of each file is distributed over the set of discs, and requests to access files arrive at the system. An overview of this problem and the potential of some methods and algorithms to address it are discussed in [14].

If a single copy of each file is distributed uniformly over $n$ discs, and $m$ files are requested at random, then the number of file requests, or load, on each disc has the same distribution as the number of balls in each bucket when $m$ balls are distributed uniformly at random
into $n$ buckets. The expected maximum load on any single disc in the system is the same as the expected maximum number of balls in any single bucket, which is well known to be $(1 + o(1)) \ln n / \ln \ln n$ with high probability when $m = n$, or $\Theta(\frac{\ln n}{\ln(1+\frac{n}{m})} + \frac{m}{n})$ with high probability for $m \geq n$ [12].

If there are multiple copies of each file then an algorithm that chooses where to allocate each request is required. A simple greedy algorithm for on-line allocation sends each file request to the least busy of the discs where the file is stored. For example, if there are two copies of each file, rather than one, randomly distributed over the $n$ discs then this algorithm yields an exponential decrease in the expected maximum load, which is $(1 + o(1)) \ln \ln n / \ln 2 + \Theta(m/n)$ with high probability [2]. Note that if there are $d$ copies, then the expected maximum load is $(1 + o(1)) \ln \ln n / \ln d + \Theta(m/n)$ with high probability. Hence the improvement from 2 to $d > 2$ copies is only linear.

Sanders, Egner and Korst [14] show that in a system where there are two copies of each file, optimal allocation can be achieved if requests are processed in batches. This is equivalent to an offline load balancing problem. Optimal allocation algorithms, however, are typically quite slow, so fast, near optimal algorithms are desirable. Sanders [35] proposes a linear time allocation algorithm in which certain discs, those with a particularly low number of requests, are prioritised for receiving allocations, and beyond that requests are allocated first to discs with the minimum load plus number of requests. It is this algorithm for offline load balancing that is analysed in this thesis.

In particular, we treat the following formulation of the offline load balancing problem: Consider a system of $n$ discs on which 2 copies of each file are uniformly distributed, and at which file requests arrive uniformly at random. For a fixed maximum load $k$, what is the maximum number of requests that Sanders’ algorithm can allocate so that the load on any disc does not exceed $k$, with high probability?
1.2 The \( k \)-core

For positive integer \( k \), the \( k \)-core of a graph, \( G \), is the largest subgraph of minimum degree at least \( k \). An example of a 3-core is shown in Figure 1.1. A \( k \)-core can be found, for example, by successive deletion of vertices of degree less than \( k \) until no more remain. At this point the graph remaining is the \( k \)-core of \( G \). If the graph remaining is empty then the \( k \)-core is empty. It’s clear that if \( G \) has a non-empty \( k \)-core, then \( G \) also has a non-empty \( l \)-core for all \( 0 < l < k \). Also, if more edges are included in \( G \) to make a new graph \( G' \), then \( G' \) also has a non-empty \( k \)-core. In other words, the existence of a \( k \)-core is an *monotone increasing* property of graphs. For random graphs, this implies that as average degree increases, so does the probability that a \( k \)-core exists.

In the uniform random graph, a property, \( Q \), is said to have a *sharply concentrated threshold*, \( c \), if, for any \( \epsilon > 0 \), the probability of \( Q \) tends to 1 when the average degree is greater than \( c + \epsilon \), and tends to 0 when the average degree is less than \( c - \epsilon \) (as the number of vertices in the random graph goes to infinity). Probabilistic analysis of algorithms that successively delete vertices of degree less than \( k \) has yielded threshold values for the existence of a \( k \)-core in a random graph, and shown that these thresholds are sharply concentrated [31].

More recently, simultaneously with the work in this thesis and in [9], the thresholds for
the $k$-core have been re-derived by several authors, using new techniques. In [30], Molloy extends a simple heuristic argument described in the preliminary material of [31] to give a short derivation of the thresholds for the appearance of cores in random $r$-uniform hypergraphs. Janson and Luczac [21] recover the main theorem of [31] using a model for random pseudographs with given degree sequence, similar to our allocation model, with a simple proof based on properties of empirical distributions of independent random variables. Kim [26] has developed a different approach to the $k$-core problem by considering a “Poisson cloning” model of a random graph. [9] covers the method used in this thesis and also includes an easy generalisation of the main result to hypergraphs.

Also recently, the size of cores in random hypergraphs with given degree sequences has been found by Cooper [11]. The result on the size of cores in Theorem 5.1 of this thesis was obtained independently, using rather different methods.

1.3 The graph model

There is a natural correspondence between the offline load balancing problem with two choices and a graph or pseudograph. As shown in Figure 1.2, let each disc be represented by a vertex and each file request by an edge. The end vertices of each edge represent the two discs where the corresponding file is stored. ‘Balancing the load’ is a matter of allocating each edge to one of its end vertices in such a way as to minimise the maximum number of edges allocated to any single vertex.

A nice way to represent the allocation is to direct each edge towards the vertex to which it is allocated. The load on a vertex is then the in-degree of the vertex, so the maximum load is the maximum in-degree in the graph. Sanders’ algorithm translates into an algorithm that generates an orientation for the edges of the graph.

We define the load-degree of a vertex $v$ (in a graph subjected to Sanders’ algorithm) to be the number of undirected edges incident with $v$ plus twice the in-degree of $v$. For example, in Figure 1.3, the edge $uv$ gets allocated to $v$ and the load-degree of $v$ increases from 3 to
Figure 1.2: A load balancing system and its representation as a graph.

4, whilst the load-degree of \( u \) decreases from 6 to 5. Sanders’ algorithm prioritises vertices for which the difference between load-degree and in-degree is no greater than \( k \), to receive allocations. We call these priority 1 vertices. If all edges incident with such a vertex, \( v \), were directed towards \( v \) then the load-degree of \( v \) would not exceed \( 2k \). If there are no priority 1 vertices then the algorithm directs an edge towards a vertex with minimum load-degree.

We ask, for fixed \( k \), in a random, \( n \) vertex, \( m \) edge graph, what is the probability that Sanders’ algorithm succeeds, that is, finds an orientation of the edges of the graph in which the maximum in-degree does not exceed \( k \)? This question is partly answered by the following.
Figure 1.3: An edge gets allocated to \( v \).

Firstly, a relationship between cores and offline load balancing with two choices was observed by Czumaj and Stemmann [12]: If a graph has no \((k + 1)\)-core then it is possible to direct all the edges in such a way that no vertex has in-degree greater than \( k \). Indeed, the priority step of Sanders’ algorithm, in effect, reduces a graph to its \((k + 1)\)-core before any edges get allocated to vertices of degree greater than \( k \). (More precisely, at some time during the algorithm the undirected part of the remaining graph is the \((k + 1)\)-core of the original graph.) So, for the uniform random graph, a.a.s. Sanders’ algorithm will succeed if the average degree of the graph is less than the threshold for the existence of a \((k + 1)\)-core.

Secondly, we have the following lemma, which follows, for example, from Hakimi’s theorem in [18]. Define a graph, \( G \), to be \( k \)-orientable if there exists an orientation of the edges of \( G \) in which no vertex has in-degree greater than \( k \).

**Lemma 1.1** A graph (or pseudograph), \( G \), is \( k \)-orientable if and only if \( G \) contains no subgraph with average degree greater than \( 2k \).

**Proof.** Suppose \( C \) is a subgraph of \( G \) with \( n \) vertices, \( m \) edges, and average degree greater than \( 2k \). Let \( \tilde{G} \) be any orientation of the edges of \( G \), and \( \tilde{C} \) the induced orientation of \( C \). By the pigeonhole principle, as \( m \geq kn + 1 \), there must be a vertex in \( \tilde{C} \), and therefore in \( \tilde{G} \), with in-degree at least \( k + 1 \). Hence the desired orientation of \( G \) does not exist.

The other direction follows from Hall’s Theorem. Assume that \( G \) contains no subgraph of average degree greater than \( 2k \). Let \( B \) be the bipartite graph with vertex classes \( X \) and
Y, with a vertex in X for each edge in G and a vertex in Y for each vertex in G. For each edge \( e = (u, v) \) in G there are edges \( (e, u) \) and \( (e, v) \) in B. Let \( H \) be any subset of X. Let \( \Gamma(H) \) denote the set of neighbours (in B) of vertices in \( H \). \( H \) corresponds to a subgraph of G with \( \Gamma(H) \) vertices and \( H \) edges. Hence

\[
|\Gamma(H)| \geq \frac{1}{k}|H|,
\]

as the average degree of any subgraph is no greater than 2k.

Now, split each vertex of Y into k nodes, each connected to \( \Gamma(y) \). In this new graph, \( B' \), \( \Gamma(H) \) is k times larger than in B, so

\[
|\Gamma(H)| \geq |H|,
\]

for any subset \( H \subseteq X \). Hence, by Hall’s Theorem, there exists a matching of size \( |X| \) in \( B' \).

Returning to B, such a matching corresponds to a subgraph, say \( M \), in which each vertex of X has degree 1 and each vertex of Y has degree no greater than k. Returning to G, this gives the desired orientation when each edge of G is directed towards the vertex to which it is connected in M. 

As an immediate corollary, if a graph, G, contains a subgraph of average degree greater than 2k then Sanders’ algorithm, or any other algorithm with the same aim applied to G cannot succeed. In particular, if G has a \( (k + 1) \)-core with average degree greater than 2k, then Sanders’ algorithm applied to G will not be successful.

Hence, in the uniform random graph, the threshold for the success of Sanders’ algorithm lies between the threshold for the existence of the \( (k + 1) \)-core, and the threshold for the graph to contain a subgraph of average degree 2k.

By showing that the threshold for the success of Sanders’ algorithm is the same as the threshold for the existence of a \( (k + 1) \)-core with average degree 2k, we show the Sanders’ algorithm is a.a.s. optimal. As a corollary (Corollary 6.2), this proves that this latter threshold is also the threshold for \( k \)-orientability, as conjectured by Karp and Saks.
Figure 1.4: A graph with minimum degree 3, average degree less than 4, but containing a subgraph of average degree greater than 4.

The contrived example of Figure 1.4 illustrates that the statements of Corollary 6.2 are only true a.a.s. and not in general.

1.4 Some notation and definitions

1.4.1 Random graphs and processes

We use the notation $[x] := \{1, 2, \ldots, x\}$ for positive integer $x$.

An $n$-vertex *pseudograph* consists of a set of vertices $V = [n]$ and a multiset of edges $E$ drawn from $\{(i, j) : (i, j) \in V \times V\}$. A pseudograph may contain loops or multiple edges.

A *multigraph* is a pseudograph which contains no loops; it consists of a set of vertices $V = [n]$ and a multiset of edges $E$ drawn from $\{(i, j) : (i, j) \in V \times V, i \neq j\}$.

A *graph*, or simple graph contains neither loops nor multiple edges, so if $E$ is the edge set of an $n$-vertex graph then $E \subseteq \{(i, j) \in V \times V : i \neq j\}$. Clearly any graph or multigraph is also a pseudograph.

By *size* of a pseudograph we mean the number of vertices, and by *density* we mean the average degree of the pseudograph.

The term *random graph* often refers to either the uniform random graph, $\mathcal{G}(n, m)$, or the binomial random graph, $\mathcal{G}(n, p)$. $\mathcal{G}(n, m)$ is the uniform probability space on $n$-vertex, $m$ edge graphs and $\mathcal{G}(n, p)$ is the probability space on all $n$-vertex graphs, with each $m$-edge
graph having probability $p^m(1 - p)^{N-m}$, where $N = \binom{n}{2}$.

In general, however, a random graph is a probability space on an underlying set of graphs, defined by a random model. A random pseudograph is an obvious generalisation: a probability space on a set of pseudographs. The random pseudograph models that we shall use are defined in Chapter 3. Each of them are defined on pseudographs with a fixed number of vertices and edges.

Note that “random graph” is also used to mean a (random) graph with the distribution of the probability space. We shall use the term “random pseudograph” in this way, and refer to our probability spaces as “probability spaces of pseudographs”. We use the symbol $\in$ as shorthand for “has/with the distribution of”, for example, if $M$ is a probability space of pseudographs, then $M \in \mathcal{M}$ means “pseudograph $M$ has the distribution of $\mathcal{M}$” or “a pseudograph $M$ with the distribution of $\mathcal{M}$”. Really, the only difference between $M$ and $\mathcal{M}$ is conceptual; it is sometimes convenient to think of the probability space as a set or space, so we use $\mathcal{M}$, and sometimes, especially when thinking about what an algorithm does, to have in mind a single pseudograph, $M$, in a black box, on which the algorithm is acting.

A random pseudograph process is also a probability space, on sequences of pseudographs indexed by time. We are only considering discrete processes, so time is assumed to be discrete in all that follows. Typically, a discrete random graph process is determined by an algorithm applied to an initial random graph. Consider an algorithm acting on a graph $G$, and let $G(0) = G$ and $G(t)$ be $G$ at time $t$. If $G$ is a random graph and the algorithm is randomised, then for each $t$, $G(t)$ has a certain distribution. Hence the sequence $\{G(t)\}_{t \geq 0}$ is a probability space with a distribution determined by $G$ and the algorithm. The random processes that we consider are each determined by a randomised algorithm applied to an initial random pseudograph, or related object, and are defined in Chapter 4. The definition of each of these algorithms guarantees that they will end by a certain time, so each process reaches a constant state in finite time. We call this constant
state the final state of the process.

1.4.2 Probability

We use \( P \) for probability and \( E \) for expectation.

The probability spaces of pseudographs and the pseudograph processes defined here are all discrete and finite so we use implicitly the \( \sigma \)-algebra consisting of all subsets of the underlying set, so that each element in the underlying set has an assigned probability.

An event is a subset of the underlying set, for example, in a probability space of pseudographs an event could be the subset of all pseudographs with a given graph property. The probability of an event is equal to the sum of probabilities of each element in the subset. We use \[ \text{“statement”} \] to mean the event described by “statement”, and we use \( I_A \) to denote the indicator variable for an event \( A \):

\[
I_A = \begin{cases} 
1 & \text{if } A \\
0 & \text{if not } A.
\end{cases}
\]

A random variable is any real function on the underlying set, for example, the number of vertices of degree \( j \) is a random variable on a space of pseudographs. Most of the random variables we define are non-negative, integer valued random variables such as this.

By restriction of a probability space, say \( \Omega \), to an event \( A \) we mean the probability space, \( \Omega_A \), which has sample space (the subset) \( A \), and probability measure induced by the probability in \( \Omega \) conditional on \( A \). That is, if \( P_\Omega \) and \( P_{\Omega_A} \) are the probability measures in \( \Omega \) and \( \Omega_A \) respectively, then for any element or event \( B \),

\[
P_{\Omega_A}(B) := \frac{P_\Omega(B \cap A)}{P_\Omega(A)} = P_\Omega(B \mid A).
\]

An object is chosen uniformly at random from a set if each object in the set has an equal probability of being chosen. We use the abbreviation u.a.r. for uniformly at random.
1.4.3 Asymptotics

Unless otherwise specified, the letter $n$ is reserved for the number of vertices, or size, of a pseudograph, which is assumed to be fixed for each probability space. Hence each random pseudograph and process is really a family of probability spaces indexed by $n$, and each event, $A$, or random variable, $X$, is really a family of events, $A_n$, or random variables, $X_n$, defined for the family of probability spaces, although the subscript will generally be omitted. Asymptotics for events and random variables shall be as $n$ goes to infinity.

An event $A_n$ defined on a family of probability spaces indexed by $n$ holds asymptotically almost surely if $P(A_n) \to 1$ as $n \to \infty$. We use the abbreviation a.a.s. for asymptotically almost surely.

We use the following definitions for asymptotic terminology, from [43]. This definition of $o()$ is non-standard, but equivalent to the usual and accommodating the a.a.s. case.

Let $f(n)$, $g(n)$ and $\phi(n)$ be functions and suppose that $|f| < \phi g$. If $\phi$ is bounded for $n$ sufficiently large we write $f = O(g)$. If $\phi \to 0$ as $n \to \infty$ we write $f = o(g)$. We write $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$, and we write $f \sim g$ if $f = (1 + o(1))g$.

We wish to make statements about sequences of random variables, involving these notations. For this purpose, for a statement $S$, we define “a.a.s. $S$” to mean that all inequalities $|f| < \phi g$ implicit in $S$ hold a.a.s.

1.4.4 The differential equation method

In [41], Wormald describes a method for the analysis of a discrete random process using differential equations, and gives a general purpose theorem for the use of this method. An algorithm which finds the $k$-core is given as an example of how this method can be applied. This method has been used to analyse several other kinds of algorithms on random graphs and random regular graphs, for example in [41] [42] [13].

The following is taken from [40]. Theorem 1.2 here is Theorem 1 of [40], incorporating Note 4 of that version. A similar, but more general version can be found as Theorem 5.1.
of [41], from which we also take Theorem 6.1, given below.

A discrete time random process, $\Omega$ is a probability space, and can be denoted by a sequence, $(Q_0, Q_1, \ldots)$ of probability spaces where each $Q_i$ takes values on some set $S$. Elements of $\Omega$ are sequences $(q_0, q_1, \ldots)$ where each $q_i$ is in $S$. $\mathcal{H}_t$ is used to denote $(Q_0, Q_1, \ldots, Q_t)$, the history of the process up to time $t$. A random variable for the process corresponds to a function defined on the histories, say $y(H_t)$, but denoted for convenience by $Y(t)$.

Now consider a sequence of random processes, $\Omega_n$, indexed by $n = 1, 2, \ldots$. There is a corresponding sequence of sets $S_n$, such that for each $n$, the elements of $\Omega_n$ are sequences $(q_0(n), q_1(n), \ldots)$ where each $q_i(n)$ is in $S_n$. In particular we are interested in the asymptotic behaviour of random variables $Y_n(t)$ defined on the histories of $\Omega_n$, as $n$ goes to infinity. For simplicity, the dependence on $n$ is usually dropped from the notation, although all asymptotics, unless otherwise stated, are for $n \to \infty$.

Let $S_n^+$ be the set of all $h_t = (q_0, q_1, \ldots, q_t)$ where each $q_i \in S_n$.

We say that a function $f(u_1, \ldots, u_j)$ satisfies a Lipschitz condition on $D \subseteq \mathbb{R}^j$ if a constant $L > 0$ exists with the property that

$$|f(u_1, \ldots, u_j) - f(v_1, \ldots, v_j)| \leq L \sum_{i=1}^j |u_i - v_i|,$$

for all $(u_1, \ldots, u_j)$ and $(v_1, \ldots, v_j)$ in $D$.

In the following, for each $l$, $1 \leq l \leq a$, $Y_l(t)$ denotes the random counterpart of a function $y_l(h_t)$. For $D \subseteq \mathbb{R}^{a+1}$, define the stopping time $T_D = T_D(Y_1, \ldots, Y_a)$ to be the minimum $t$ such that $(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \notin D$.

**Theorem 1.2** Let $a$ be fixed. For $1 \leq l \leq a$, let $y_l : \cup_n S_n^+ \to \mathbb{R}$ and $f_l : \mathbb{R}^{a+1} \to \mathbb{R}$, such that for some constant $C_0$ and all $l$, $|y_l(h_t)| < C_0 n$ for all $h_t \in S_n^+$ for all $n$. Suppose also that, for some bounded connected open set $D \subseteq \mathbb{R}^{a+1}$ containing the intersection of $\{(t, z_1, \ldots, z_a) : t \geq 0\}$ with some neighbourhood of $\{(0, z_1, \ldots, z_a) : \Pr(Y_l(0) = z_l n, 1 \leq l \leq a) \neq 0 \text{ for some } n\}$,

the following three conditions hold:

$$
\{$$
(i) There is a constant $C_1$ such that for all $l$,
\[ |Y_l(t+1) - Y_l(t)| \leq C_1, \]
for all $t < T_D$;

(ii) For all $l$,
\[ \mathbb{E}(Y_l(t+1) - Y_l(t)|\mathcal{H}_t) = f_l(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) + o(1) \]
for all $t < T_D$;

(iii) For each $l$ the function $f_l$ is continuous and satisfies a Lipschitz condition on $D$.

Then:

(a) For $(0, \hat{z}_1, \ldots, \hat{z}_a) \in D$ the system of differential equations
\[ \frac{dz_l}{dx} = f_l(x, z_1, \ldots, z_a), \quad l = 1, \ldots, a \]
has a unique solution in $D$ for $z_l : \mathbb{R} \to \mathbb{R}$ passing through
\[ z_l(0) = \hat{z}_l, \quad l = 1, \ldots, a \]
and which extends to points arbitrarily close to the boundary of $D$;

(b) Almost surely
\[ Y_l(t) = nz_l(t/n) + o(n) \]
uniformly for $0 \leq t \leq \min\{\sigma n, T_D\}$ and for each $l$, where $z_l(x)$ is the solution in (a) with $\hat{z}_l = Y_l(0)/n$, and $\sigma = \sigma(n)$ is the supremum of those $x$ to which the solution can be extended.

We use also the following variation of Theorem 1.2, which is Theorem 6.1 of [41] (we have changed the statement of the theorem slightly to refer to Theorem 1.2, rather than a variation of that theorem, Theorem 5.1 of [41].) In this version $D$ is replaced by $\hat{D}$, and a larger domain $D$ is defined. The theorem states that, so long as condition (iii) holds in $D$, the solution to the system of equations can be extended beyond the boundary of $\hat{D}$, into $D$. 

19
Theorem 1.3 For any set $\hat{D} = \hat{D}(n) \subseteq \mathbb{R}^{a+1}$, define the stopping time $T_{\hat{D}} = T_{\hat{D}(n)}(Y_1, \ldots, Y_a)$ to be the minimum $t$ such that $(t/n, Y_1(t)/n, \ldots, Y_a(t)/n) \notin \hat{D}$. Assume that the first two hypotheses of Theorem 1.2 apply only with the restricted range $t < T_{\hat{D}}$ of $t$. Then the conclusions of the theorem hold as before, except with $0 \leq t \leq \min\{\sigma n, T_{\hat{D}}\}$ replaced by $0 \leq t \leq \min\{\sigma n, T_{\hat{D}}\}$.

$\hat{D}$ can be defined so that the boundary of $\hat{D}$, $\partial \hat{D}$, is a natural boundary for the random variables, for example, the values of the variables at $\partial \hat{D}$ indicate the final state the process. Then, if a larger set $D \supset \hat{D}$ can be defined, in which (iii) remains valid, it follows that a.a.s. $T_{\hat{D}} < \sigma n$ and by Theorem 1.3 the continuous approximation is valid up to time $T_{\hat{D}}$, which represents a natural ending time for the process.

1.4.5 Branching processes

Some variables in the processes we analyse have near zero expectation, and applying the differential equation method to find continuous approximations for these variables would require some careful justification. Instead, for such variables we can sometimes define a related branching process, such that the expected size or the expected number of children in the branching process bounds the expectation of the variable in question. We analyse this branching process to deduce information about the distribution of the random variable. This method of analysis can be found in [36] and [39].

A Galton-Watson process is a branching process in which the number of new children in each birth step is independently and identically distributed. Formally, given a probability measure $\mu$ on $\mathbb{N} := \{0, 1, 2, \ldots\}$, the associated Galton-Watson process is a Markov chain $(N_k, k \geq 0)$ with $N_0 = 1$ and values in $\mathbb{N}$ such that, conditionally on $N_{n-1}$, for each $n \geq 1$

$$N_n = \sum_{i=1}^{N_{n-1}} X_{i,n},$$

where variables $X_{i,n}$ are independently, identically distributed with distribution $\mu$.

Say that a non-negative, integer valued random variable $W$ is stochastically dominated by
random variable $\tilde{W}$ if

$$P(W \geq r) \leq P(\tilde{W} \geq r) \quad \text{for } r \geq 0.$$

In general, the branching processes we consider are not Galton-Watson processes as the distribution of the number of children, say $X_t$, in each birth step varies. We get around this by finding a random variable, $Z$, which stochastically dominates $X_t$ for each $t$ (or at least for the first $\log^2 n$ or so steps of the process), and hence the expected size of the branching process with $X_t$ children in each step is bounded by the expected size of the Galton-Watson process with $Z$ children in each step.

It’s well known that if $E(Z) \leq 1$ then the size of the Galton-Watson process is finite with probability 1. We use the following, stronger result on the size of a Galton-Watson process in some of our arguments. This theorem and proof is taken from [39, Theorem 3.2] and was essentially shown by Crâmer in the 1920s.

Say that a non-negative, integer valued random variable $Y$ has a $K, c$ tail if for all non-negative integers $u$,

$$P(Y \geq u) \leq Ke^{-cu}.$$

If $Y$ has a $K, c$ tail for some $K$ and $c$, we say that $Y$ has an exponentially small tail.

**Theorem 1.4** Let $K, c$ be positive reals. Let $Z$ be a non-negative integer valued random variable with a $K, c$ tail and with $\exp(Z) = \mu < 1$. Let $X$ be the size of the Galton-Watson branching process in which each node, independently, has $Z$ children. Then there exist positive $K^+, c^+$, dependent on $\mu$, $K$, $c$, such that $X$ has a $K^+, c^+$ tail.

**Proof.** Fix any positive $\lambda < c$, say $\lambda = \frac{c}{2}$ for definiteness. The Laplace Transform $E[e^{tZ}]$ is then defined for all $0 \leq t \leq \lambda$. For such $t$ we also have $E[(Z-1)^2e^{tZ}] \leq E[(Z-1)^2e^{\lambda Z}]$ which is bounded by a convergent sum. Let $M$ be an explicit upper bound on $E[(Z-1)^2e^{tZ}]$. Now, using the standard association of branching processes with random walks, we have

$$P[X \geq s + 1] \leq P[Z_1 + \ldots + Z_s \geq s] \quad (1.1)$$

21
where the $Z_i$ (number of children of the $i$-th node) are independent, each with distribution $Z$. For any $0 \leq t \leq \lambda$ we have the Chernoff bound

$$P[Z_1 + \ldots + Z_s \geq s] \leq E[e^{t(Z_1+\ldots+Z_s)}]e^{-ts} = E[e^{tZ}e^{-ts}]^s.$$ 

Let us set

$$\phi(t) := e^{-t}E[e^{tZ}] = E[e^{t(Z-1)}].$$

We have $\phi(0) = 1$ and

$$\phi'(0) = E[Z - 1] = \mu - 1$$

which, critically, is negative. Also,

$$\phi''(t) = E[(Z - 1)^2e^{t(Z-1)}] \leq M$$

for all $t \in [0, \lambda]$. Hence

$$\phi(t) \leq 1 + (\mu - 1)t + (1 + M)\frac{t^2}{2}$$

for all $t \in [0, \lambda]$. We set $t = \frac{1-\mu}{1+M}$ if this value is less than $\lambda$, otherwise we set $t = \lambda$. Either way we get an explicit $U < 1$ and the bound $\phi(t) \leq U$. This gives Theorem 1.4 with $K^+ = U^{-1}$ and $c^+ = -\ln U$. 

### 1.5 Organisation of thesis

In Chapter 2, we give the load balancing and $k$-core algorithms, $\textbf{Core}(k)$ and $\textbf{Load}(k)$, that we shall analyse.

In Chapter 3 we describe the allocation model for random pseudographs and show that the distribution of degrees of vertices in this model is multinomial and asymptotically Poisson. Then we define a conditioned allocation model for pseudographs with the degree sequence of low degree vertices specified. We show that in this restricted space, the distribution of degrees of high degree vertices is restricted multinomial and asymptotically truncated Poisson. Finally, we show the connection between pseudographs and the uniform random graph.
In Chapter 4, the random pseudograph processes determined by \textbf{Core}(k) and \textbf{Load}(k) are described. First an edge deletion step and an edge allocation step for a pseudograph are defined. Then we show that in a pseudograph subjected to either of these steps the distribution of high degree vertices remains restricted multinomial. This enables us to analyse the \( k \)-core and load balancing processes as Markov chains in which the states are restricted spaces of pseudographs, and the transition probabilities are determined by a simple set of parameters for each restricted space. To do this we define two simpler allocation objects that we call partial pre-allocations, and two algorithms, \textbf{ACore}(k) and \textbf{ALoad}(k), for partial pre-allocations. These determine two new random processes that we use to analyse \textbf{Core}(k) and \textbf{Load}(k). \textbf{ACore}(k) and \textbf{ALoad}(k) dynamically generate and delete (or orient) the pairs, or edges, of an allocation, as well as some of the degree sequence.

In Chapter 5, the \( k \)-core process is analysed using the differential equation method, and thresholds for the existence of a \( k \)-core are derived. In Chapters 6 and 7, the load balancing process is analysed and it is shown that the load balancing algorithm is asymptotically almost surely optimal. In Chapter 8 some other algorithms and related problems are discussed.

Chapter 9 contains material on the development of heuristic algorithms for the Euclidean Steiner tree problem.

There are three appendices. In the first the proof of a lemma from Chapter 6 is given. The second is an index of symbols and notation and the third, an index of terminology.
Chapter 2

The algorithms for pseudographs

This chapter is devoted to the description and definition of two specific algorithms. The first algorithm finds, for fixed \( k > 0 \), the \( k \)-core of a pseudograph, if it exists. It does this by deleting, one at a time, edges incident with vertices of degree less than \( k \), until no more remain. Isolated vertices are also deleted. The final pseudograph is the \( k \)-core of the original pseudograph.

The second algorithm represents Sanders’ algorithm for an off-line load balancing problem with two choices and fixed maximum load \( k \). This algorithm attempts to allocate each edge of a pseudograph to one of its end vertices in such a way that the maximum number of allocations to any single vertex is no greater than \( k \). Allocation is represented by directing each edge towards the vertex to which it is allocated and the algorithm is successful if the maximum in-degree of vertices in the final pseudograph does not exceed \( k \).

The motivation for analysing the algorithm on pseudographs rather than simple graphs is twofold. Firstly, for the load balancing application, a pseudograph, or perhaps a multigraph, is a more appropriate model for a system of \( n \) discs containing a large number of files; if there are no multiple edges then this is equivalent to insisting that no two files are saved on the same pair of discs, which restricts the total number of files to \( n^2/2 \).

Secondly, consider a random graph process that, from a starting graph \( G_0 \), repeatedly
deletes edges chosen u.a.r. Let $E_{v_1v_2}(i)$ be the event that the $i$th edge deleted is $v_1v_2$. If $G_0 \in \mathcal{G}(n, m)$, then $E_{v_1v_2}(i)$ is dependent on the events $E_{v_1v_2}(j)$ for each $j < i$, and also on the event that $v_1$ is distinct from $v_2$. On the other hand, if $G_0$ is a random pseudograph (with the distribution we shall define in the next chapter) then the events $E_{v_1v_2}(i)$ are independent of each other because multiple edges are allowed. This independence simplifies the analysis somewhat. Moreover, a lemma of Chvátal’s [10], given here as Lemma 3.11, shows that properties true a.a.s. for pseudographs are also true a.a.s. for graphs and, incidentally, multigraphs.

### 2.1 Algorithm for the $k$-core

Let $M$ be a pseudograph and $k > 0$ a fixed integer. The first algorithm reduces $M$ to its $k$-core:

**Algorithm 2.1 Core($k$) for $M$**

$$V_k = \{\text{vertices of degree less than } k\}$$

Repeat the following:

1. select a vertex $v \in V_k$ u.a.r.
2. select an edge incident to $v$ u.a.r. and delete it
3. If $v$ is isolated
   - delete $v$
   - update $V_k$

Until $V_k = \emptyset$ Output the remaining pseudograph

Let $M_c$ be the output of Core($k$). Then if $M_c$ is not empty it is the $k$-core of $M$. Note that if $M$ is a simple graph, then $M_c$ is also, since no new edges are introduced. Hence, Core($k$) can be used to find the $k$-core of a graph.
In **Core**($k$), the vertex and edge to be deleted are selected u.a.r., however, they could alternatively be selected by any rule without affecting the outcome of the algorithm. The $k$-core is determined by the graph and not the algorithm and so is independent of any such rule and the order in which vertices are deleted. Here, we specify that the vertex to be deleted is chosen uniformly at random, since this makes for a nice probabilistic analysis of the underlying random process.

### 2.2 Algorithm for load balancing

Let $k$ be a fixed integer. The second algorithm attempts to direct all the edges of a pseudograph, $M$, so that no vertex has in-degree greater than $k$. A prioritisation scheme ensures that vertices of a certain degree and in-degree are processed before others of higher degree. As a consequence, at some point during the algorithm, the unoriented part of the remaining pseudograph will be the $(k + 1)$-core of $M$. This is explained further after the definition of the algorithm.

Define the **load-degree** of a vertex, $v$ to be the number of undirected edges incident with $v$ plus twice the in-degree of $v$ and let $V_{d,j}$ denote the set of vertices with load-degree $d$ and in-degree $j$. In the algorithm, **allocate to $v$** means “Select an undirected edge incident with $v$ u.a.r. and direct this edge towards $v$.”

It is convenient to assign to each vertex a **priority** that is a function of load-degree and in-degree. If $v$ is a vertex in $V_{d,j}$ with $d - j \leq k$ and $d - 2j > 0$, then the remaining $d - 2j$ edges incident with $v$ may be allocated to $v$ without exceeding the maximum load. After allocating all remaining edges, $v$ has load-degree $j'$ with

$$j' = j + (d - 2j) = d - j \leq k.$$

Such vertices are treated as vertices with the highest priority, and shall be called **priority 1 vertices**.

Priority 1 vertices are chosen first for processing. When there are no priority 1 vertices, a vertex of minimum load-degree is chosen for processing.
Algorithm 2.2 Load($k$) for $M$

$V_P = \{\text{priority 1 vertices}\}$

$V_{2k} = \{\text{vertices } v \in V_{d,j} : d \leq 2k \text{ and } d - 2j > 0\}$

Repeat the following:

If $(V_P \neq \emptyset)$

- select $v \in V_P$ u.a.r.
- allocate to $v$

Else

- select a vertex $v \in V_{2k}$ of minimum load-degree u.a.r.
- allocate to $v$

update $V_P$ and $V_{2k}$

Until $V_{2k} = \emptyset$ Output the remaining pseudograph

End

Let $M_f$ be the output of Load($k$). Note that $M_f$ is dependent on the order in which vertices of $M$ are processed, whereas $M_c$ is not. Say that Load($k$) has a successful output if $M_f$ has no undirected edges and maximum in-degree $k$. Note that, as for Core($k$), if $M$ is simple, then so is $M_f$ so Load($k$) is also an algorithm for load balancing on a graph.

By a step of Core($k$) or Load($k$) we mean one iteration of the repeat loop, which corresponds either to the deletion or the allocation of one edge.

Priority 1 vertices are chosen first for processing by Load($k$), so in an “initial phase”, the algorithm only allocates edges to vertices in $V_P$, until $V_P$ is empty. Firstly, note that when an edge is allocated to a priority 1 vertex, $v$, if there is still an undirected edge incident with $v$ after the allocation, then $v$ is still a priority 1 vertex. As a consequence, vertices remain in $V_P$ until all their incident edges have been allocated. Secondly, the vertices starting in, or entering $V_P$ in this initial phase are exactly those which, at some time, have in-degree zero and load-degree $k$ or less. These two things would apply if the algorithm deleted edges in the initial phase, rather than orienting them, and $V_P$ would be the same set of vertices. In this case, the vertices in $V_P$ would be those of degree $k$ or less. By
comparison with \textbf{Core}(k + 1) and the definition of \( V_{k+1} \) in that algorithm, we see that at the end of the initial phase, when \( V_P \) first becomes empty, the undirected part of the remaining pseudograph is exactly the \((k + 1)\)-core of the initial pseudograph.
Chapter 3

The conditioned pseudograph model

In this chapter we define random pseudographs and show some useful lemmas regarding their properties. These results will be used in the subsequent chapters. We will be interested in random pseudographs for which the sequence of low degree vertices is specified. First we define an allocation model for random pseudographs and show some properties of its degree sequence. Then we define a conditioned allocation model for pseudographs with a specified sequence of low degree vertices (less than a fixed integer). We show that in this model the distribution of degrees of high degree vertices is restricted multinomial and asymptotically truncated Poisson. Finally, we give Chvátal’s lemma, which shows that properties that hold a.a.s. for pseudographs also hold a.a.s. for simple graphs.

3.1 The allocation model

An allocation of $2m$ balls into $n$ buckets is defined as follows:

Take a set of $2m$ balls arranged in $m$ pairs and throw each ball sequentially into one of $n$ buckets chosen u.a.r. Call this an allocation of $2m$ balls into $n$ buckets. From each
allocation we obtain a pseudograph, with each bucket representing a vertex, each ball a point in a vertex and each pair of balls forming an edge. More formally:

**Definition 3.1** An allocation is a function \( f : [l] \to [n] \). If \( l = 2m \) then a pseudograph with \( m \) edges on vertex set \([n]\) is obtained by including an edge between vertices \( i \) and \( j \) whenever \( f(2r - 1) = i \) and \( f(2r) = j \), for \( 1 \leq r \leq m \). All functions are equiprobable.

Let \( A(n, l) \) denote the (uniform) probability space of allocations and let \( M(n, m) \) denote the probability space of pseudographs determined by \( A(n, 2m) \).

This model was used by Bollobás and Frieze [7], and Chvátal [10]. Note that \( M(n, m) \) is not a uniform probability space of pseudographs: there are \( 2^m m! \) allocations giving each simple graph but, for example, only half as many giving each graph with exactly one double edge (think of the number of permutations of the labels on the pairs of balls which give the same graph). However, as we state in Section 3.3, it follows that \( M(n, m) \) contains a uniform copy of the uniform random graph, \( G(n, m) \). Hence, we conclude using Lemma 3.11 that what is true a.a.s. for pseudographs in \( M(n, m) \) is true a.a.s. for random simple graphs.

The use of pseudographs instead of simple graphs affords a number of simplifications, mainly because each edge \((i, j)\) is uniformly distributed on \([n] \times [n]\), independently of every other edge. This simplifies enumeration formulae, the distributions of vertex degrees, and the transition probabilities for edge deletion and similar processes. In Chapter 4 we define random pseudograph processes where, in each time step, the location of a single edge is “revealed” and that edge is “processed” by being either deleted or assigned an orientation. We exploit the fact that, on choosing the location of one endpoint of the edge, the location of the other endpoint is uniformly distributed on all possibly locations. With a simple graph model this is not exactly true; there can be no multiple edges so the location of the second endpoint has some dependence on the history of the process and the location of edges that have already been processed (in practice, this means an error term needs to be included in the transition probabilities for the process).
We will analyse properties of pseudographs with reference to degree sequences. For convenience we define some spaces of non-negative integer valued vectors, which are sets of valid degree sequences, or partial degree sequences, subject to certain constraints. The first such space is the set of degree sequences of length \( d \) and with terms summing to \( s \).

For even \( s \), this space contains all possible degree sequences for pseudographs on \( d \) vertices with \( s/2 \) edges.

Let \( D^d_s := \{(h_1, \ldots, h_d) : \sum_{i=1}^{d} h_i = s\} \)

where each term \( h_i \) is a non-negative integer.

Let \( M \) be a pseudograph with the distribution of \( \mathcal{M}(n, m) \) and let \( D(M) = (D_1, \ldots, D_n) \) be the degree sequence of \( M \). Its joint distribution is multinomial. For \( \mathbf{d} = (d_1, \ldots, d_n) \in D^d_{2m} \),

\[
P(\mathbf{D}(M) = \mathbf{d}) = n^{-2m}(2m)!/\prod_{i=1}^{n} d_i!,
\]

(3.1)

since \((2m)!/\prod d_i!\) is the number of allocations with degree sequence \((d_1, \ldots, d_n)\) and \(n^{2m}\) is the total number of allocations.

On the other hand, let \( Z(\lambda) \) be a Poisson variable with mean \( \lambda \). For fixed \( \lambda > 0 \) let \( \mathbf{Z} = (Z_1, \ldots, Z_n) \) be a vector of \( n \) independent copies \( Z_i \) of \( Z(\lambda) \).
For $z = (z_1, \ldots, z_n) \in D_{2m}^n$,

$$P(Z = z) = \prod_{i=1}^{n} P(Z_i = z_i)$$

$$= \prod_{i=1}^{n} e^{-\lambda z_i} / z_i!$$

$$= e^{-\lambda n} \lambda^{2m} / \prod z_i!,$$

(3.2)

and thus $D(M)$ has the same distribution as $Z$ restricted to the event $D_{2m}^n$.

In fact, for $m = O(n)$, as $n \to \infty$, each $D_i$ is asymptotically Poisson distributed. For non-negative integer $j$,

$$P(D_i = j) \sim \lambda^j e^{-\lambda} / j!,$$

(3.3)

for $\lambda = E(D_i) = 2m/n$. This is shown as a particular case in Lemma 3.6.

Let $X_j$ be the number of vertices of degree $j$ in $M$. Then

$$E(X_j) = \sum_{i=1}^{n} P(D_i = j)$$

$$\sim n \frac{(2m/n)^j}{e^{2m/n} j!}.$$

As shown in Lemma 1 of [9], the $X_j$ are sharply concentrated as $n \to \infty$. We will see similar expressions to that on the right hand side in Chapter 4, where we consider certain probabilities determined by the numbers of vertices of each degree in a pseudograph, and again in Chapters 5 and 6 where we shall define random variables based on the numbers of vertices of each degree.

### 3.2 The conditioned allocation model

Now we define a restriction of $M(n, m)$ to pseudographs with a given degree sequence of low degree vertices, and investigate its properties.

We define a second space of vectors, $L_d^1$, with non-negative integer valued terms. This space contains low (less than $l$) degree sequences for pseudographs with $d$ vertices of degree less than $l$. 32
For integers \( l > 0 \) and \( d \geq 0 \) let

\[
L^d_l := \{(h_1, \ldots, h_d) : 0 \leq h_i < l, i = 1, \ldots, d\},
\]

where each term \( h_i \) is a non-negative integer.

By the low degree sequence of a pseudograph we mean, for a given \( l \), the shortened degree sequence obtained from the degree sequence by removing all the terms greater than or equal to \( l \). In other words, for a pseudograph with \( n \) vertices in total and \( d \) vertices of degree less than \( l \), the low degree sequence is the (ordered) list of degrees of these \( d \) vertices.

We shall not be concerned which of the \( n \) vertices are those of low degree and in our definitions we shall only mention pseudographs in which the low degree sequence is an initial segment of the degree sequence: the low degree vertices are the first \( d \) vertices.

By doing this we are assuming an implicit understanding that the restriction of \( \mathcal{M}(n, m) \) to pseudographs with a given low degree sequence has the same distribution whichever subset of \([n]\) the low degree vertices are. This shall be explained in more detail after the definition of the restricted space of pseudographs.

Let \( d \in L^*_l \). For a pseudograph \( M \), let \( \mathcal{D}_d \) be the event that each vertex \( i \) of \( M \) has degree \( d_i \) for \( 1 \leq i \leq r \) and degree greater than or equal to \( l \) for \( i \geq r + 1 \),

\[
\mathcal{D}_d := \{ d(i) = d_i, \ 1 \leq i \leq r \text{ and } d(i) \geq l, \ i \geq r + 1 \}.
\]

Clearly, for \( M \) with the distribution of \( \mathcal{M}(n, m) \), \( \mathcal{D}_d \) is non-empty if and only if \( n - r \geq 0 \) and \( 2m - \sum_i d_i \geq (n - r)l \).

**Definition 3.2** For integers \( l > 0 \), \( r \geq 0 \) and \( d \in L^*_l \), define the probability space of pseudographs,

\[
\mathcal{M}_l(d, \nu, s),
\]

to be the restriction of \( \mathcal{M}(n, m) \) to \( \mathcal{D}_d \), where \( \nu = n - r \) and \( s = 2m - \sum_{i=1}^{r} d_i \).
Hence, \( M \in \mathcal{M}_l(d, \nu, s) \) has \( \nu \) vertices of degrees each greater than or equal to \( l \) and summing to \( s \). These are the last \( \nu \) vertices: vertices \( r+1 \) to \( n \).

Further to the discussion above, we could define a more general restricted space where we also specify a subset \( R \in [n] \) of size \( r \) and restrict to pseudographs with low degree vertices \( R \) having low degree sequence \( d \in L_r \). However, to avoid unnecessarily complicating the notation and ensuing discussion, we do not. Indeed, given any such \( R \), the vertices can be relabelled canonically so that the low degree vertices are labelled 1 to \( r \) and the other vertices are labelled \( r+1 \) to \( n \) (simply preserve the order within each list). This permutation of vertex labels clearly does not affect the distribution of the restricted space.

Hence, in what follows, whenever we refer to \( \mathcal{M}_l(d, \nu, s) \) we understand implicitly that our arguments also apply with the same conclusion to the more generally defined restricted space with low degree vertices in a set \( R \neq [r] \). Although the latter is what we would need on some occasions to be absolutely precise, we shall only ever refer to restricted pseudographs in which the low degree vertices are the first few, knowing that our arguments are easily made precise by either specifying the set of low degree vertices or performing the canonical relabelling described above.

As before, let \( Z(\lambda) \) be a Poisson variable with mean \( \lambda \), and let \( Z^{(l)}(\lambda) \) be the restriction of \( Z(\lambda) \) to \( Z \geq l \). So, for integer \( j \),

\[
P(Z^{(l)}(\lambda) = j) = \begin{cases} \frac{\lambda^j}{j!} & j \geq l \\ 0 & \text{otherwise} \end{cases}
\]

where, for integer \( a > 0 \), \( f_a(\lambda) \) is a truncation of the exponential series:

\[
f_a(\lambda) := \sum_{i \geq a} \frac{\lambda^i}{i!}.
\]

For convenience we set \( f_a(\lambda) = f_0(\lambda) = e^\lambda \) for all \( a < 0 \). Note that for any integer \( a \),

\[
f'_a(\lambda) = \sum_{i \geq \max(a,1)} \frac{i \lambda^{i-1}}{i!} = f_{a-1}(\lambda).
\]

Let

\[
\psi_l(\lambda) := \frac{\lambda f_{l-1}(\lambda)}{f_l(\lambda)},
\]

\[34\]
so $\psi_l(\lambda)$ is the mean of $Z^{(l)}(\lambda)$.

For fixed $\lambda > 0$ let

$$Z^{(l,d)} := (Z^{(l)}_1, \ldots, Z^{(l)}_d),$$

a vector of $d$ independent copies $Z^{(l)}_i$ of $Z^{(l)}(\lambda)$.

We define another space of vectors, $H^{d}_{l,s}$, with non-negative integer valued terms. This space contains all high ($l$ or greater) degree sequences for pseudographs with $d$ vertices of degree $l$ or greater, and with these degrees summing to $s$. The high degree sequence of a pseudograph is obtained by removing all the terms less than $l$ from the degree sequence.

Let

$$H^{d}_{l,s} := \{(h_1, \ldots, h_d) : \sum_{i=1}^{d} h_i = s \text{ and } h_i \geq l\},$$

where each term $h_i$ is a non-negative integer. So, $H^{\nu}_{l,s}$ contains all high degree sequences for pseudographs in $\mathcal{M}_l(d, \nu, s)$, for any $d \in L^r_l$.

**Definition 3.3** Define the restricted multinomial distribution, $\text{Multi}(l, d, s)$ to be the restriction of the multinomial distribution to $H^{d}_{l,s}$. That is, for $b = (b_1, \ldots, b_d)$,

$$P(b) = \begin{cases} \frac{A}{\prod_{i=1}^{d} b_i!} & b \in H^{d}_{l,s} \\ 0 & \text{otherwise,} \end{cases}$$

where

$$A = \left( \sum_{h \in H^{d}_{l,s}} \frac{1}{\prod_{i=1}^{d} h_i!} \right)^{-1}.$$  

The following lemma gives results analogous to (3.1) and (3.2) for pseudographs in $\mathcal{M}_l(d, \nu, s)$.

**Lemma 3.4** For integers $l > 0$, $r \geq 0$ and $d \in L^r_l$, let $M$ be a pseudograph with the distribution of $\mathcal{M}_l(d, \nu, s)$ and let $D_l(M) = (D_1, \ldots, D_\nu)$ be the degree sequence of $M$, ignoring the vertices of degree less than $l$ (and relabelling the vertices accordingly).

The joint distribution of $D_l(M)$ is $\text{Multi}(l, \nu, s)$, and consequently $D_l(M)$ has the same distribution as $Z^{(l,\nu)}$ restricted to the event $H^{\nu}_{l,s}$.
**Proof.** Let \( n = r + \nu \) and \( 2m = s + \sum_i d_i \). Firstly, for any \( h \in H_{i,s}^\nu \), the number, \( A_d \), of allocations in \( A(n, 2m) \) giving pseudographs with degree sequence \( \vec{d} = (d, h) \) is

\[
A_d = \frac{(2m)!}{\prod_{i=1}^r d_i! \prod_{i=1}^\nu h_i!}.
\]

Let \( A_{d,s,\nu} \) be the total number of allocations which give pseudographs in \( M_l(d, \nu, s) \). Then

\[
\mathbf{P}(D_l(M) = h) = \frac{A_d}{A_{d,s,\nu}} = B_{d,s,\nu} \prod_{i=1}^\nu \frac{1}{h_i!},
\]

where \( B_{d,s,\nu} \) is constant. Hence \( D_l(M) \) has the distribution of \( \text{Multi}(l, \nu, s) \).

Secondly,

\[
\mathbf{P}(Z_{l,\nu} = h) = \prod_{i=1}^\nu \frac{\lambda^{h_i}}{f_i(\lambda) h_i!} = \frac{\lambda^s}{f_l(\lambda)^\nu} \prod_{i=1}^\nu h_i!,
\]

and thus \( D_l(M) \) has the same distribution as \( Z_{l,\nu} \) restricted to the event \( H_{i,s}^\nu \).  

Next, we give some properties of the truncated Poisson distributed random variable \( Z_{l,\nu}(\lambda) \).

For any \( l \geq 0 \), \( Z_{l,\nu}(\lambda) \) has mean given by

\[
\mathbf{E}(Z_{l,\nu}(\lambda)) = \sum_j j \mathbf{P}(Z_{l,\nu}(\lambda) = j)
\]

\[
= \sum_{j \geq l} j \frac{\lambda^j}{f_l(\lambda) j!}
\]

\[
= \frac{\lambda}{f_l(\lambda)} \sum_{j \geq \max(l,1)} \frac{\lambda^{j-1}}{(j-1)!}
\]

\[
= \frac{\lambda f_{l-1}(\lambda)}{f_l(\lambda)} = \psi_l(\lambda),
\]

(3.7)
Let \( \psi = \psi_l(\lambda) \). To determine the variance of \( Z = Z(l)(\lambda) \),

\[
\begin{align*}
\mathbf{E}(Z(Z - 1)) &= \sum_i i \mathbf{P}(Z(Z - 1) = i) \\
&= \sum_j j(j - 1) \mathbf{P}(Z(Z - 1) = j(j - 1)) \\
&= \sum_j j(j - 1) \mathbf{P}(Z = j) \\
&= \sum_{j \geq \max(l,2)} j(j - 1) \frac{\lambda^j}{f_l(\lambda)j!} \\
&= \frac{\lambda^2 f_{l-2}(\lambda)}{f_l(\lambda)} \\
&= \eta \psi,
\end{align*}
\]

where \( \eta = \eta_l(\lambda) := \psi_{l-1}(\lambda) \). So

\[
\text{Var}(Z) = \mathbf{E}(Z(Z - 1)) + \mathbf{E}(Z) - \mathbf{E}(Z)^2
\]

\[
= \eta \psi + \psi - \psi^2
\]

\[
= \psi(\eta + 1 - \psi), \quad (3.8)
\]

as shown in [32]. The variance of \( Z \) must be positive for positive \( \lambda \), since \( \mathbf{P}(Z \neq \psi) \) is positive.

In the next lemma some properties of the function \( \psi_l(\lambda) \) are shown.

**Lemma 3.5** For fixed \( l \geq 0 \) and \( \lambda \in (0, \infty) \),

(i) \( \lim_{\lambda \to \infty} \psi_l(\lambda) = \infty \),

(ii) \( \lim_{\lambda \to 0^+} \psi_l(\lambda) = l \),

(iii) \( \psi_l(\lambda) \geq \lambda \) with \( \psi_l(\lambda) = \lambda \) if and only if \( l = 0 \),

(iv) \( \psi_l(\lambda) < l + \lambda \left( \frac{\lambda + 1}{l + \lambda + 1} \right) \),

(v) \( \psi_l(\lambda) \) is monotonically increasing with \( \lambda \).

(vi) \( \psi_{l+1}(\lambda) > \psi_l(\lambda) \).
Proof. \( \psi_l(\lambda) \) may be rearranged to be

\[
\psi_l(\lambda) = \lambda + \frac{l}{1 + \sum_{j \geq 1} \frac{\lambda^j}{(l+j)!}} \tag{3.9}
\]

\[
< \lambda + \frac{l}{1 + \frac{\lambda}{l+1}}
\]

\[
= l + \frac{\lambda(1 + \lambda)}{l + \lambda + 1},
\]

from which (i), (ii), (iii) and (iv) are clear, with equality in (iii) only for \( l = 0 \) as \( l/(1 + \frac{\lambda}{l+1}) \) increases with \( l \). To show (v) we observe, as in [31] and [32], that

\[
\frac{d\psi_l(\lambda)}{d\lambda} = \frac{f_{l-1}(\lambda)}{f_l(\lambda)} + \frac{\lambda f_{l-2}(\lambda)}{f_l(\lambda)} - \frac{\lambda f_{l-1}(\lambda)^2}{f_l(\lambda)^2}
\]

\[
= \frac{1}{\lambda} (\psi + \psi \eta - \psi^2)
\]

\[
= \frac{1}{\lambda} \text{Var}(Z^{[l]}(\lambda))
\]

\[
> 0,
\]

for \( \lambda > 0 \).

To show (vi), by (3.9)

\[
\psi_{l+1}(\lambda) - \psi_l(\lambda) = \frac{1}{l+1} + \sum_{j \geq 1} \frac{\lambda^j}{(l+1+j)!} - \frac{1}{l} + \sum_{j \geq 1} \frac{\lambda^{j-1}}{(l+j)!}.
\]

Now \( \frac{1}{l} > \frac{1}{l+1} \) and for each \( j \geq 1 \)

\[
\frac{\lambda^j}{(l+j)!} > \frac{\lambda^j}{(l+1+j)!}
\]

so \( \psi_{l+1}(\lambda) - \psi_l(\lambda) > 0 \). □

For \( c > l \) let \( \lambda_{l,c} \) denote the positive root of the equation

\[
\psi_l(\lambda) = c. \tag{3.10}
\]

Equivalently, \( \lambda_{l,c} \) is the Poisson parameter required for \( Z^{[l]}(\lambda) \) to have mean \( c \). It follows from Lemma 3.5 (i), (ii) and (v) that \( \lambda_{l,c} \) exists uniquely for all \( c > l \).

Set

\[
p_{l,c,j} := \frac{\lambda_{l,c}^j}{f_l(\lambda_{l,c})j!} \tag{3.11}
\]

38
Then, by (3.4), for \( j \geq l \), \( p_{l,c,j} \) is the probability that \( Z^{(l)}(\lambda_{l,c}) \) is equal to \( j \).

Note that if \( l = 0 \), then \( \lambda = c = \psi_{0}(\lambda) = \eta_{0}(\lambda) \), so \( E(Z^{(0)}(\lambda)) = \lambda \) and \( \text{Var}(Z^{(0)}(\lambda)) = \lambda \) by (3.7) and (3.8).

**Lemma 3.6** Let \( l \geq 0 \) and \( c > l \) be fixed. Let \( M_{\nu} \) be a pseudograph with the distribution of \( M_{l}(d, \nu, cv) \) and let \( D_{l}(M_{\nu}) = (D_{1}, \ldots, D_{\nu}) \) be the degree sequence of \( M_{\nu} \), ignoring the vertices of degree less than \( l \). Then the \( D_{i} \) are (individually) asymptotically truncated Poisson distributed as \( \nu \to \infty \),

\[
P(D_{\nu} = j) \sim \begin{cases} p_{l,c,j} & j \geq l \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}
\]

**Proof.** If \( l = 0 \) then \( M_{\nu} \) has the distribution of \( M(\nu, cv) \) and the lemma implies (3.3). The following proof extends quite naturally to this case and the case \( l = 1 \), using \( f_{k}(\lambda) = f_{0}(\lambda) = e^{\lambda} \) for all \( k < 0 \).

Let \( s = cv \) and let \( \lambda = \lambda_{l,c} \) be fixed in all the following. Note that \( \lambda_{l,c} \) exists since \( c > l \).

Let \( Z^{(l,\nu)} = (Z_{1}, \ldots, Z_{\nu}) \) be a vector of \( \nu \) independent copies of \( Z^{(l)}(\lambda) \), and similarly define \( Z^{(l,\nu-1)} \). By Lemma 3.4, for \( j \geq l \),

\[
P(D_{\nu} = j) = P(Z_{\nu} = j \mid Z^{(l,\nu)} \in H_{l,s}^{\nu})
= \frac{P(Z_{\nu} = j \wedge Z^{(l,\nu)} \in H_{l,s}^{\nu})}{P(Z^{(l,\nu)} \in H_{l,s}^{\nu})}
= \frac{P(Z_{\nu} = j \wedge Z^{(l,\nu-1)} \in H_{l,s-j}^{\nu-1})}{P(Z^{(l,\nu)} \in H_{l,s}^{\nu})}
= \frac{P(Z_{\nu} = j)P(Z^{(l,\nu-1)} \in H_{l,s-j}^{\nu-1})}{P(Z^{(l,\nu)} \in H_{l,s}^{\nu})}
= \frac{\lambda^{j}P(Z^{(l,\nu-1)} \in H_{l,s-j}^{\nu-1})}{f_{l}(\lambda)j!P(Z^{(l,\nu)} \in H_{l,s}^{\nu})}
= \frac{\lambda^{j}P(\sum_{i=1}^{\nu-1} Z_{i} = s - j)}{f_{l}(\lambda)j!P(\sum_{i=1}^{\nu} Z_{i} = s)}.
\]

Let

\[
R(\nu) = \frac{P(\sum_{i=1}^{\nu-1} Z_{i} = s - j)}{P(\sum_{i=1}^{\nu} Z_{i} = s)}. \tag{3.13}
\]
To show that \( R(\nu) \) is asymptotic to 1, we follow the proof of Theorem 4, part (a) in [32], filling in the details not given there.

Let \( a_n(k) = P(\sum_{i=1}^{n} Z_i = k) \). Then \( R(n) = \frac{a_n-1(cn-j)}{a_n(cn)} \). The asymptotic behaviour of \( a_n(k) \) can be determined by applying the following lemma of Bender, Lemma 2 of [4].

In this lemma the notation \([r]\) refers to the integer part of \( r \).

**Lemma 3.7** (Bender) If \( a_n(k) \) is log concave and asymptotically normal with mean \( \mu_n \) and variance \( \sigma_n^2 \) with \( \sigma_n \to \infty \), then for \( S = (-\infty, \infty) \),

\[
\lim_{n \to \infty} \sup_{x \in S} \left| \sigma_n p_n([\sigma_n x + \mu_n]) - \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \right| = 0, \tag{3.13}
\]

where \( p_n(k) = \frac{a_n(k)}{\sum_{j} a_n(j)} \). Furthermore, the rate of convergence depends only on \( \sigma_n \) and the rate of convergence of \( a_n(k) \) to asymptotic normality.

First, asymptotic normality of \( a_n(k) \) is established by the Berry-Esseen inequality (3.14).

The following theorem is taken from [15].

**Theorem 3.8** (Berry-Esseen) Let \( X_k \) be independent random variables with distribution \( F \), such that

\[
E(X_k) = 0, \quad E(X_k^2) = \sigma^2 > 0, \quad E(|X_k|^3) = \rho < \infty.
\]

Let \( F_n \) be the distribution of the normalised sum,

\[
\sum_{i=1}^{n} \frac{X_i}{\sigma \sqrt{n}}.
\]

Then for all \( x \) and \( n \),

\[
\left| F_n(x) - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy \right| \leq \frac{3\rho}{\sigma^3 \sqrt{n}}. \tag{3.14}
\]

Let \( Z = Z^{(l)}(\lambda) \), so \( E(Z) = c \) by (3.7) and (3.10). Let \( X_k, k = 1, 2, \ldots \) be independent copies of \( (Z - c) \). Then \( E(X_k) = 0 \) and \( \text{Var}(X_k) = \text{Var}(Z) = c(\eta + 1 - c) \) by (3.8), where \( \eta = \psi_{\lambda-1}(\lambda) \). Let \( \sigma^2 = c(\eta + 1 - c) \) be the variance of \( X_k \). Note \( \sigma \) is positive for positive \( \lambda \).
Similarly, we find that

\[ \mathbf{E}(X_k^3) = \gamma \eta c + c(1 - c)(1 - 2c + 3\eta), \]

with \( \gamma = \psi_{r-2}(\lambda) \). Let \( \rho = \mathbf{E}(|X_k|^3) = |\mathbf{E}(X_k^2)| \), and note that \( \rho \) is bounded.

Let \( F_n \) be as described in Theorem 3.8 and let \( Y_x \subseteq (-\infty, x] \) be the set of values of \( y \), no greater than \( x \), for which \( \mathbf{P}(\sum_{i=1}^{n} \frac{X_i}{\sigma} = y) \) is non-zero. \( Y_x \) is countable for any \( x \) because \( Z_i \) is integer valued and \( X_i = Z_i - c \). We have,

\[
\begin{align*}
F_n(x) &= \sum_{y \in Y_x} \mathbf{P}(\sum_{i=1}^{n} \frac{X_i}{\sigma} = y) \\
&= \sum_{y \in Y_x} \mathbf{P}(\sum_{i=1}^{n} X_i = \sigma \sqrt{ny}) \\
&= \sum_{y \in Y_x} \mathbf{P}(\sum_{i=1}^{n} Z_i = \sigma \sqrt{ny} + nc) \\
&= \sum_{k \leq \sigma \sqrt{nx} + nc} \mathbf{P}(\sum_{i=1}^{n} Z_i = k) \\
&= \sum_{k \leq \sigma \sqrt{nx} + nc} a_n(k).
\end{align*}
\]

Hence, by (3.14) \( a_n(k) \) satisfies a central limit theorem:

\[
\left| \sum_{k \leq \sigma \sqrt{nx} + nc} a_n(k) - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy \right| \leq \frac{3\rho}{\sigma^3 \sqrt{n}} \to 0 \quad \text{as } n \to \infty.
\]

That is, \( a_n(k) \) is asymptotically normal with mean \( \mu_n = nc \) and variance \( \sigma_n^2 = \sigma^2 n \). Note that \( \sigma_n \to \infty \) as \( n \to \infty \).

A sequence \( \{b_k\} \) is log concave if for all \( m \),

\[ b_m^2 \geq b_{m-1} b_{m+1}. \]

Next, to show log concavity of \( a_n(k) \), we apply the following lemma which is, in essence, the case \( r = 2 \) of Theorem 1.2 in [22, page 394], and is quite well known in general. Some more details are given in the proof here.
Lemma 3.9 The convolution of two log concave sequences with non-negative terms is log concave.

Proof. Let $\{a_n\}$ and $\{b_n\}$ be two log concave sequences with non-negative terms. Assume that all terms $a_k = b_k = 0$ for all $k < 0$ are defined. Let $\{c_n\}$ be the convolution of $\{a_n\}$ and $\{b_n\}$:

$$c_n = \sum_i a_i b_{n-i} = \begin{cases} \sum_{i=0}^n a_i b_{n-i} & n \geq 0 \\ 0 & n < 0. \end{cases}$$

Log concavity of $a_n$ implies that for all integers $m, r$ and $s$,

$$a_m a_{m+s-r} - a_{m-r} a_{m+s} \geq 0,$$

and similarly for $b_n$. Clearly this is also a sufficient condition for log concavity since the case $s = r = 1$ is included.

Let $A$ and $B$ be the infinite upper triangular matrices $A(i, j) = a_{j-i}, i, j \geq 1$ and $B(i, j) = b_{j-i}, i, j \geq 1$. Log concavity of $\{a_n\}$ is equivalent to all the order 2 minors of $A$ being non-negative, because, for any integers $i, j, r, s > 0$,

$$A \begin{pmatrix} i & i+r \\ j & j+s \end{pmatrix} = \begin{vmatrix} A(i, j) & A(i, j+s) \\ A(i+r, j) & A(i+r, j+s) \end{vmatrix} = \begin{vmatrix} a_{j-i} & a_{j+s-i} \\ a_{j-i-r} & a_{j+s-i-r} \end{vmatrix} = a_{j-i} a_{j+s-i-r} - a_{j-i-r} a_{j+s-i} = a_m a_{m+s-r} - a_{m-r} a_{m+s},$$

on substituting $m = j - i$.

Let $C = AB$, that is,

$$C(i, j) = \sum_{k \geq 1} A(i, k) B(k, j) = \sum_k a_{k-i} b_{j-k} = \sum_t a_t b_{j-i-t} = c_{j-i}.$$
So $C$ is well defined and is the corresponding upper triangular matrix for $c_n$. Thus $c_n$ is log concave if and only if all the order 2 minors of $C$ are non-negative. Since $C = AB$, the minors of $C$ are a sum of products of the minors of $A$ and $B$,

$$C \left( \begin{array}{cc} i, & i + r \\ j, & j + s \end{array} \right) = \sum_{k<l} A \left( \begin{array}{cc} i, & i + r \\ k, & l \end{array} \right) B \left( \begin{array}{cc} k, & l \\ j, & j + s \end{array} \right),$$

which is the Cauchy-Binet formula applied to matrices of infinite order, and can be seen directly by expanding all the terms. This sum must be finite since each $C(i,j)$ is, and each minor of $C$ must be positive since the minors of $A$ and $B$ are. Hence all the order 2 minors of $C$ are non-negative and $c_n$ is log concave.

Now, let $b_k = P(Z^{(l)}(\lambda) = k)$. The sequence $\{b_k\}$ is log concave, as, for $i \leq l$, $b_{i-1}b_{i+1} = 0 \leq b_i^2$, and for $i > l$,

$$b_{i-1}b_{i+1} = \frac{\lambda^{i-1}}{f_l(\lambda)(i-1)!} \frac{\lambda^{i+1}}{f_l(\lambda)(i+1)!} \frac{\lambda^{2i}}{f_l(\lambda)^2(i-1)!(i+1)!}$$

$$= \frac{b_i^2 i}{i + 1} \leq b_i^2.$$

Note that $a_1(k) = b_k$ by definition, and $a_n(k)$ is the convolution of $a_{n-1}(k)$ and $b_k$:

$$a_n(k) = \sum_{i=0}^{k} P(\sum_{j=1}^{n-1} Z_j = i)P(Z_n = k - i)$$

$$= \sum_{i=0}^{k} a_{n-1}(i)b_{k-i}.$$

Hence, by Lemma 3.9, $a_2(k)$ is log concave and, by induction, $a_n(k)$ is log concave for all $n$.

Finally, applying (3.13) and noting that $\sum_i a_n(i) = 1$ so $p_n(k) = a_n(k)$ in this case,

$$\sigma_n a_n([\sigma_n x + \mu_n]) \sim \frac{1}{\sqrt{2\pi}} e^{-x^2/2},$$
for all $x \in (-\infty, \infty)$. That is,

$$a_n(k) \sim \frac{1}{\sigma_n \sqrt{2\pi}} e^{-\frac{(k-\mu_n)^2}{2\sigma_n^2}},$$

for all integers $k$.

In particular,

$$a_n(cn) \sim \frac{1}{\sigma \sqrt{2\pi n}},$$

and

$$a_n(cn - j) \sim \frac{1}{\sigma \sqrt{2\pi (n-1)}} e^{-\frac{(c-j)^2}{2\sigma^2(n-1)}}.$$

Hence

$$R(n) = \sqrt{n} \frac{\sqrt{n}}{\sqrt{n-1}} e^{-\frac{(c-j)^2}{2\sigma^2(n-1)}} (1 + o(1)) \sim 1.$$

This completes the proof. 

Let

$$m_{l,d,s,j} := P(h_1 = j \mid (h_1, \ldots, h_d) \in \text{Multi}(l, d, s)).$$

(3.15)

In this corollary, properties of the degree sequence of pseudographs in the conditioned allocation model are summarised.

**Corollary 3.10** Let $M$ be a pseudograph with the distribution of $\mathcal{M}_l(d, \nu, s)$ and let $D_l(M) = (D_1, \ldots, D_\nu)$ be the degree sequence of $M$, ignoring the vertices of degree less than $l$. Then for any $1 \leq i \leq \nu$ and $j \geq l$,

$$P(D_i = j) = m_{l,i,s,j}$$

(3.16)

and, as $\nu \to \infty$,

$$m_{l,i,s,j} \sim p_{l,\nu,s,j}.$$ 

(3.17)

**Proof.** Firstly, by Lemma 3.4, $D_l(M)$ has the distribution of $\text{Multi}(l, \nu, s)$, so (3.16) follows. Secondly, (3.17) is immediate from (3.16) and Lemma 3.6. 

44
3.3 Simple graphs

Let \( N = \binom{n}{2} \). Then \( \binom{N}{m} \) is the number of graphs on vertex set \([n]\) with \( m \) edges, \( m!2^m \) is the number of allocations that give rise to each simple graph (by permuting the labels on the pairs of balls), and \( n^{2m} \) is the total number of allocations possible. This leads to the following lemma, used in [31, page 125] and taken from [10].

**Lemma 3.11** (Chvátal) For \( M \) with the distribution of \( \mathcal{M}(n, m) \) and \( c = 2m/n \),

\[
\Pr(M \text{ is simple}) = \frac{N \cdot m!2^m}{n^{2m}} \sim \exp(-c/2 - c^2/4).
\]

The important consequence of this lemma is that what we show to be true a.a.s. for pseudographs in \( \mathcal{M}(n, m) \), is true a.a.s. for random simple graphs.

**Corollary 3.12** For \( G \in \mathcal{G}(n, m) \), \( M \in \mathcal{M}(n, m) \) and property \( Q \), if \( M \) has \( Q \) a.a.s then \( G \) has \( Q \) a.a.s.

**Proof.** Let \( A, B \) and \( C \) be the following events:

\[
A_n = [M \text{ has property } Q], \\
B_n = [M \text{ is simple}], \\
C_n = [G \text{ has property } Q].
\]

Lemma 3.11 shows that for fixed \( n \) and \( m \), \( \Pr(B_n) \) is constant, which implies that the restriction of \( \mathcal{M}(n, m) \) to simple graphs is uniform and hence has the distribution of \( \mathcal{G}(n, m) \). Hence

\[
\Pr(C_n) = \Pr(A_n | B_n) \\
= 1 - \Pr(\neg A_n | B_n) \\
= 1 - \frac{\Pr(\neg A_n \wedge B_n)}{\Pr(B_n)} \\
\geq 1 - \frac{\Pr(\neg A_n)}{\Pr(B_n)}.
\]
\( \mathbf{P}(B_n) \) is asymptotically constant, while \( \mathbf{P}(\neg A_n) \to 0 \) as \( n \to \infty \), hence \( \mathbf{P}(C_n) \to 1 \) and

\( G \) has property \( Q \) a.a.s. \( \blacksquare \)
Chapter 4

The pseudograph processes

Here, restricted space refers specifically to a restriction of $\mathcal{M}(n, m)$ to $\mathcal{D}_d$, for some low degree sequence $d$, as per Definition 3.2, or to a similarly defined restricted space of pseudographs with some oriented edges. The latter are given in Definition 4.9.

In this chapter we define two random processes by which we shall analyse the algorithms $\text{Core}(k)$ and $\text{Load}(k)$ defined in Chapter 2, and we give functions for the transition probabilities between the states of these processes. Note that we use the letter $k$ as the parameter for both algorithms: We have “$k$-core” and “maximum load $k$”. Recall that $\text{Load}(k)$ mirrors $\text{Core}(k + 1)$ in an initial phase, except that it orients, rather than deletes edges, until the unoriented part of the pseudograph is the $(k+1)$-core of the original pseudograph. So, in this context, will sometimes refer to $(k+1)$-cores, rather than $k$-cores. We will see that $\text{Load}(k)$ is also related to $\text{Core}(2k + 1)$, so some objects will have $2k+1$, rather than $k$, as a parameter.

Let $\Omega_{\text{Core}(k)}$ and $\Omega_{\text{Load}(k)}$ be the random processes determined by $\text{Core}(k)$ and $\text{Load}(k)$ applied to $M \in \mathcal{M}(n, m)$. It’s easy to see from the definitions of the algorithms that the processes they determine are Markov chains; the distribution of a pseudograph after a step of either algorithm depends only on the current pseudograph and not on any prior pseudograph. We analyse the Markov chains by considering the single edge deletion and
edge orientation steps on which \textbf{Core}(k) and \textbf{Load}(k) are based. These are specified in Section 4.1. The edges of a pseudograph are given by the pairing of points in the allocation that determines the pseudograph. In Section 4.2, Lemma 4.2 we show that for \( M \in \mathcal{M}(n, m) \) this pairing is uniformly distributed on all pairings of \([2m]\). As a consequence, we will be able to define equivalent edge deletion and edge orientation steps that generate the paring for each edge before deleting or orienting it. We will ultimately define algorithms based around these steps that act on pseudographs where each edge remains random until immediately before it is deleted or oriented.

In Section 4.3 we consider the effect of a single step on the distribution of a pseudograph and, in Lemma 4.3 and Corollary 4.5, we show that the truncated multinomial distribution of high degree vertices is maintained. This means that at any time, \( M \) has the distribution of some restricted space. We exploit this and rather than treat \( \Omega_{\text{Core}(k)} \) and \( \Omega_{\text{Load}(k)} \) as Markov chains on individual pseudographs, we model them as Markov chains in which the states are the restricted spaces of pseudographs.

First, we consider \( \Omega_{\text{Core}(k)} \). In Section 4.4, we analyse the transition probabilities between states in the chain. The transition probability between any two restricted spaces, \( \mathcal{M}_1 \) and \( \mathcal{M}_2 \), with respect to the edge deletion step, is defined to be the probability that \( M' \) has the distribution of \( \mathcal{M}_2 \), where \( M' \) is the result of applying the edge deletion step to \( M \in \mathcal{M}_1 \).

In Lemma 4.8 we show that the transition probability between any two restricted spaces, say \( \mathcal{M}_k(d, \nu, s) \) and \( \mathcal{M}'_k(d', \nu', s') \), is a function of \( d, \nu, s, d', \nu', s' \) only. Hence, we can simply use the vectors \((d, \nu, s)\) for the states in the Markov chain modelling \( \Omega_{\text{Core}(k)} \).

In Section 4.5 we define a simpler allocation object, the partial pre-allocation, which is essentially just the vector \((d, \nu, s)\). A partial pre-allocation represents a partly determined pseudograph, part of which is fixed, and part of which is random: The pairing of the points is random, and the degrees of high degree vertices are random. Each partial pre-allocation corresponds uniquely to a restricted space. In Section 4.6, we define two algorithms, \( \text{ACore}(k) \) and \( \text{ALoad}(k) \), that act on partial pre-allocations by, in each step, generating part of the allocation, generating a pairing of two points and deleting or orienting that pair.
of points (that edge). These determine the two processes, $\Omega_{\text{ACore}}(k)$ and $\Omega_{\text{ALoad}}(k)$, that we shall ultimately use, instead of $\Omega_{\text{Core}}(k)$ and $\Omega_{\text{Load}}(k)$, to analyse $\text{Core}(k)$ and $\text{Load}(k)$. Our motivation for this is, when it comes to the analysis, we avoid having to justify why certain events in the pseudograph processes have certain probabilities, because, by the definition of $\text{ACore}(k)$ and $\text{ALoad}(k)$, the probabilities of these events are obvious for each step of $\Omega_{\text{ACore}}(k)$ and $\Omega_{\text{ALoad}}(k)$. Moreover, in Lemmas 4.23 and 4.25 of Section 4.5, we show that the outputs of $\text{Core}(k)$ and $\text{Load}(k)$ are evident from $\Omega_{\text{ACore}}(k)$ and $\Omega_{\text{ALoad}}(k)$.

### 4.1 Edge deletion and orientation steps

In the allocation and conditioned allocation models of pseudographs that we use, a pseudograph consists of labelled vertices, each containing a number of labelled points, and a canonical pairing of the points, $\{(2i-1, 2i), i = 1, \ldots, m\}$, which determines the $m$ edges of the pseudograph. The two objects, a pair in the underlying pairing and the corresponding edge in the pseudograph, shall be treated as equivalent. Any reference to an edge is understood to be a reference to a pair of points in the pairing, and visa-versa. The mate of a point is the point to which it is paired in the underlying pairing.

In Section 4.2 we show that the labels of the points, aside from determining the pairing, have no bearing on the distribution of the pseudograph. Moreover, any uniformly random pairing of the points in a random pseudograph determines a pseudograph with the same distribution as for the canonical pairing. This follows from the same principle we used to justify our definition of the restricted space of pseudographs, $\mathcal{M}_l(d, \nu, s)$, on page 33. There, we argued that a permutation of the labels of the vertices does not change the distribution of a pseudograph, and consequently it is sufficient to only consider restricted spaces of pseudographs in which the low degree sequence is an initial segment of the degree sequence. In Lemma 4.2 we apply the same idea to the labels of the points in the vertices and argue that a permutation does not change the distribution of the random pseudograph. Consequently, we allow ourselves to regard the points in our pseudographs as unlabelled and paired via a uniformly random pairing.
The degree, \( d(v) \), of a vertex \( v \) is the number of points in \( v \). In Load\((k)\), edges are given an orientation. If \( e_1e_2 \) is an edge of the pseudograph that is oriented towards \( e_1 \), call \( e_1 \) an \textit{in-point} and \( e_2 \) an \textit{out-point}. Call a point which is not an in-point or an out-point a \textit{free point}. The \textit{load-degree}, \( d^L(v) \), of a vertex \( v \) is the number of free points in \( v \) plus twice the number of in-points in \( v \). The \textit{in-degree}, \( d^-(v) \), of a vertex \( v \) is the number of in-points in \( v \). Call a pseudograph in which some edges are oriented, a \textit{partially oriented pseudograph}.

Load\((k)\) prioritises vertices of lowest load-degree. For fixed \( k \), \textit{priority} is defined as a function of load-degree and in-degree, \((d, j)\), as follows.

\textbf{Definition 4.1} For fixed \( k \), the function \( \text{Pri}_k \) is given by

\[
\text{Pri}_k(d, j) = \begin{cases} 
1 & \text{if } d - j \leq k \text{ and } d - 2j > 0 \\
\infty & \text{if } d - 2j = 0 \\
d - k + 1 & \text{otherwise.}
\end{cases}
\]

A vertex, \( v \), has priority \( r \) with respect to (w.r.t.) \( k \) if

\[
\text{Pri}_k(v) := \text{Pri}_k(d^L(v), d^-(v)) = r.
\]

A vertex has \textit{highest priority} when its priority has the \textit{lowest} value. The vertices with priority 1 (priority 1 vertices) have the highest possible priority and are those in which the number of in-points will not exceed \( k \) even if all remaining free points in the vertex become in-points.

Core\((k)\) is based around the following edge deletion step:

\textbf{EdgeDel}(k)

Choose a vertex \( v \) of degree \( 0 < d(v) < k \) u.a.r.

Choose a point \( e_1 \) in \( v \).

Delete \( e_1 \) and its mate \( e_2 \).

Load\((k)\) is based around the following edge orientation step:
Choose a vertex \( v \) of highest priority w.r.t. \( k \), u.a.r.

If \( \text{Pri}_k(v) \leq k + 1 \),

Choose a point \( e_1 \) in \( v \).

Orient the corresponding edge \( e_1e_2 \) towards \( e_1 \).

### 4.2 Random pairings

A pairing, \( P \) on the set \([2m]\) is a perfect matching of the points in \([2m]\). An allocation, \( f : [2m] \to [n] \) and a pairing, \( P \) of \([2m]\) together determine a pseudograph \( M_P \) on \([n]\) by including an edge between vertices \( i \) and \( j \) whenever there is a pair \((p,q)\) in \( P \) such that \( f(p) = i \) and \( f(q) = j \). Further to the discussion in Section 4.1, the following lemma justifies why we can regard the points in our pseudographs as unlabelled and paired via any uniformly random pairing.

**Lemma 4.2** Let \( f : [2m] \to [n] \) be a random allocation, and let \( M_0 \) be the pseudograph determined by \( f \) in the allocation model. Let \( P \) be a uniformly random pairing on \([2m]\) and let \( M_P \) be the pseudograph determined by \( f \) and \( P \). Then \( M_P \) has the same distribution as \( M_0 \).

**Proof.** The distribution of \( M_0 \) is determined by the uniform probability space of allocations \( \mathcal{A}(n,m) \) where a pseudograph is obtained from an allocation and the canonical pairing, \( P_0 = \{(2i-1,2i), i = 1, \ldots, m\} \).

A uniformly random pairing \( P \) can be determined by randomly permuting the labels of the points in \( P_0 \) (there are \( m!2^m \) permutations which determine each pairing). Hence a pseudograph, \( M_\sigma \) with the same distribution as \( M_P \) can be determined by composing a random permutation \( \sigma \) of \([2m]\) with a random allocation \( f : [2m] \to [n] \).

For any allocation \( g : [2m] \to [n] \), \( P(f_\sigma = g) = P(f = g) = n^{-2m} \) and hence \( f_\sigma \) is uniformly distributed on \( \mathcal{A}(n,m) \). It follows that \( M_\sigma \) and consequently, \( M_P \) have the
same distribution as $M_0$. \qed

Note that, as a corollary, if we restrict to the event that $M_0$ has low degree sequence $\mathbf{d}$, the same conclusion applies. Namely, that $M_0$ and $M_P$ both have the distribution of $\mathcal{M}_l(\mathbf{d}, \nu, s)$, for some $\nu$ and $s$, since changing the pairing does not change the degree sequence or the number of vertices of degree $l$ or greater.

For our purposes, the important consequence of the lemma is that if $M$ is a pseudograph with the distribution of $\mathcal{M}(n, m)$ or a restricted space, $\mathcal{M}_l(\mathbf{d}, \nu, s)$, then the pairing of points in $M$ is uniformly distributed on all possible pairings. This means that if $e$ is a point in $M$, then the mate of $e$ is uniformly distributed on all other points in $M$.

### 4.3 Heavy vertices

In the context of $\text{EdgeDel}(k)$, vertices of degree $k$ or greater shall be called **heavy vertices**. $\text{EdgeDel}(k)$ chooses vertices of degree less than $k$ for processing, so heavy vertices are not processed by $\text{EdgeDel}(k)$. Points in heavy vertices may be deleted, but only as mates of points that are chosen for deletion.

In the context of $\text{EdgeOri}(k)$, heavy vertices shall be those of load-degree greater than $2k + 1$, and those of load-degree $2k + 1$ with in-degree zero. These vertices have priority at least $k + 2$ w.r.t. $k$, so are not processed by $\text{EdgeOri}(k)$.

There are a few points to note regarding heavy vertices that, if not immediately clear, shall be clear by the end of this chapter. Consider a pseudograph to which $\text{EdgeDel}(k)$ or $\text{EdgeOri}(k)$ is applied and the points $e_1$ and $e_2$ in the selected edge. Firstly, $e_1$, when chosen, is not in a heavy vertex, so heavy vertices can never contain any in-points. Secondly, if $e_2$ is in a heavy vertex, then, as a corollary to Lemma 4.2, deletion of $e_2$ (as in $\text{EdgeDel}(k)$) is equivalent to the deletion of a randomly chosen point in a heavy vertex. Similarly, orienting $e_1e_2$ (as in $\text{EdgeOri}(k)$) is equivalent to changing a randomly chosen free point to an out-point. Thirdly, since out-points do not contribute to the load-degree or affect the processing of $\text{EdgeOri}(k)$, the distribution of the load-degree sequence of heavy
vertices after an application of \textbf{EdgeOri}(k), is the same as the distribution of the degree sequence of heavy vertices after an application of \textbf{EdgeDel}(k) to the same pseudograph. In other words, if out-points are disregarded, the effect of \textbf{EdgeDel}(k) and \textbf{EdgeOri}(k) on heavy vertices is the same. Finally, the degree or load-degree of a heavy vertex can never increase.

In the next lemma we show that when a pseudograph with the distribution of \( M_k(d, \nu, s) \) is subjected to \textbf{EdgeDel}(k), the heavy vertices maintain a multinomial distribution. The corollary shows that the same applies to partially oriented pseudographs subjected to \textbf{EdgeOri}(k).

Recall the set \( L_r^k \) of low degree sequences,

\[
L_r^k := \{ h = (h_1, \ldots, h_r) : 0 \leq h_i < l, \ i = 1, \ldots, r \}.
\]

For \( d \in L_r^k \), we say that a pseudograph has \textit{low degree sequence} \( d \) if each vertex \( i \) has degree \( d(i) = d_i \) for \( 1 \leq i \leq r \), and \( d(i) \geq k \) for \( i > r \). (As discussed on page 34, probability spaces of pseudographs that are the same modulo a permutation of the labels of the vertices are equivalent for purposes here, thus we need only consider pseudographs for which the low degree sequence is always an initial segment of the degree sequence.) Vertices \( i > r \) are heavy vertices with regard to \textbf{EdgeDel}(k).

Similarly, we define a set \( \hat{L}_r^k \) of low degree sequences for partially oriented pseudographs. Let

\[
\hat{L}_r^k := \{ (h, j) = (h_1, \ldots, h_r, j_1, \ldots, j_r) : 0 \leq h_i \leq l, \ 0 \leq j_i \leq \left\lfloor \frac{h_i}{2} \right\rfloor \wedge h_i - 2j_i < l, \ i = 1, \ldots, r \}.
\]

For \((d, j) \in \hat{L}_{2k+1}^r\), say that a partially oriented pseudograph has \textit{low load-degree sequence} \((d, j)\) if each vertex \( i \) has load-degree \( d_L^L(i) = d_i \) and in-degree \( d^-\L(i) = j_i \) for \( 1 \leq i \leq r \), and \( d_L^L(i) \geq 2k + 1 \) and \( d^-\L(i) = 0 \) for \( i > r \). Vertices \( i > r \) are heavy vertices with regard to \textbf{EdgeOri}(k).

For a pseudograph \( M \), and \( d \in L_k^r \), let \( D_{d, \nu, s} \) be the event that \( M \) has low degree sequence
and an additional $\nu$ vertices of degree $k$ or greater containing a total of $s$ points:

$$D_{d,\nu,s} := [d(i) = d_i, 1 \leq i \leq r \land d(i) \geq k, i \geq r + 1 \land \sum_{i=r+1}^{\nu} d(i) = s].$$

Similarly, for a partially oriented pseudograph $M$, and $(d, j) \in \hat{L}^{r}_{2k+1}$, let $\hat{D}_{d,j,\nu,s}$ be the event that $M$ has low load-degree sequence $(d, j)$ and an additional $\nu$ vertices of load-degree $2k + 1$ or greater containing a total of $s$ free points and no in-points:

$$\hat{D}_{d,j,\nu,s} := [d^L(i) = d_i, d^-(i) = j_i, 1 \leq i \leq r \land d^L(i) \geq 2k + 1, d^-(i) = 0, i \geq r + 1 \land \sum_{i=r+1}^{\nu} d^L(i) = s].$$

**Lemma 4.3** For any $k \geq 2$, $d \in L^r_k$, $d' \in L^r_k$ and non-negative $s$, $\nu$, $s'$, $\nu'$, let $M$ be a pseudograph with the distribution of $\mathcal{M}_k(d, \nu, s)$. Apply $\text{EdgeDel}(k)$ to $M$ to get $M'$. Then, restricting $M'$ to $D_{d',\nu',s'}$, $M'$ has the distribution of $\mathcal{M}_k(d', s', \nu')$.

**Proof.** Consider an allocation giving rise to $M$ in the allocation model of pseudographs. The deletion step $\text{EdgeDel}(k)$ is equivalent by symmetry to selecting the last pair of points in the allocation and deleting it, conditioned on the last point being in a vertex of degree less than $k$. In the event $D_{d',\nu',s'}$, deleting the last pair of points results in an allocation generated exactly as in $\mathcal{A}(n, 2m - 2)$ restricted to $D_{d',\nu',s'}$, where $n = r + \nu$ and $2m = s + \sum_i d_i$. Hence $M'$ has the distribution of $\mathcal{M}_k(d', s', \nu')$ as required. \qed

**Definition 4.4** Define the unoriented part, $\text{Unor}(M)$, of a partially oriented pseudograph $M$ to be the subgraph obtained when all the oriented edges of $M$ are deleted.

Note that if $e$ is the degree sequence of the heavy vertices of $\text{Unor}(M)$, then the load-degree sequence of the heavy vertices of $M$ is $(e, 0)$.

**Corollary 4.5** For any $k \geq 2$, $(d, j) \in \hat{L}^r_{2k+1}$, $(d', j') \in \hat{L}^r_{2k+1}$ and non-negative $s$, $\nu$, $s'$, $\nu'$, let $M$ be a pseudograph with low load-degree sequence $(d, j)$ and unoriented part,
Unor(M) with the distribution of $M_{2k+1}(d - 2j, \nu, s)$. Apply \textbf{EdgeOri}(k) to M to get $M'$. Let $\hat{D}$ be the event $\hat{D}_{d', \nu', s'}$ for $M'$. Then, restricting to $\hat{D}$, Unor($M'$) has the distribution of $M_{2k+1}(d' - 2j', s', \nu')$.

\textbf{Proof.} Unor($M'$) can be obtained from Unor($M$) by deleting the edge $e_1e_2$. Let $\hat{D}$ be the event $\hat{D}_{d - 2j, \nu, s}$ for Unor($M'$). Clearly $\hat{D}$ implies $D$ so the result follows directly from Lemma 4.3.

Let $M_0$ be a pseudograph with the distribution of $M(n, m)$, and let $M_t$ be the pseudograph after \textbf{EdgeDel}(k) has been applied $t$ times. Assume that $M_{t+1} = M_t$ if \textbf{EdgeDel}(k) cannot be applied to $M_t$, that is if $M_t$ has low degree sequence 0. Thus, $\{M_t\}_{t \geq 0}$ is the probability space determined by \textbf{EdgeDel}(k), $n$ and $m$, and $\{M_t\}_{t \geq 0} = \Omega_{\text{Core}(k)}$. Each element of $\{M_t\}_{t \geq 0}$ is a sequence of pseudographs.

By Lemma 4.3 applied inductively, for any $d \in L_k$, $s$ and $\nu$, restricting to $D_{d, \nu, s}$, $M_t$ has the distribution of $M_k(d, \nu, s)$.

Similarly, for some $r_0$, $d^{(0)} \in L_{2k+1}^r$ and $s_0$ let $N_0$ be a pseudograph with the distribution of $M_{2k+1}(d^{(0)} - r_0, s_0)$, and let $N_t$ be the partially oriented pseudograph after \textbf{EdgeOri}(k) has been applied $t$ times to $N_0$. Assume that $N_{t+1} = N_t$ if \textbf{EdgeOri}(k) cannot be applied to $N_t$, that is if $N_t$ has low load-degree sequence $(d, j)$ with $d_i - 2j_i = 0$ or $d_i = 2k + 1$ for each $i$. Thus, $\{N_t\}_{t \geq 0}$ is the probability space determined by \textbf{EdgeOri}(k), $n$, $d_0$ and $s_0$. Each element of $\{N_t\}_{t \geq 0}$ is a sequence of partially oriented pseudographs. $\{N_t\}_{t \geq 0}$ is not the same as $\Omega_{\text{Load}(k)}$, as $N_0$ does not have the distribution of $M(n, m)$, but if $N_0$ has no vertices of degree $k$ or less then $\{N_t\}_{t \geq 0}$ is contained in the restriction of $\Omega_{\text{Load}(k)}$ to pseudographs with $(k + 1)$-core $N_0$.

Note that \textbf{EdgeOri}(k) can choose vertices up to load-degree $2k$ for processing and when a point in a vertex of load-degree $2k$ becomes an in-point, then the load-degree of that vertex increases to $2k + 1$ and the in-degree is positive. So the low load-degree sequence of $N_t$ is always in $\hat{L}_{2k+1}^r$ and the low degree sequence of Unor($N_t$) is always in $L_{2k+1}^r$, for some $r$, $r_0 \leq r \leq n$.}
In fact, by Corollary 4.5 applied inductively, for any \((d, j) \in \hat{L}_{2k+1}\), \(s\) and \(\nu\), restricting \(N_t\) to \(\hat{D}_{d, j, \nu, s}\), \(\text{Unor}(N_t)\) has the distribution of \(\mathcal{M}_{2k+1}(d - 2j, \nu, s)\).

In particular, this means when a random pseudograph is subjected to either algorithm, \textbf{Core}(\(k\)) or \textbf{Load}(\(k\)), the multinomial distribution of the numbers of heavy vertices of each degree is maintained in each step. In the next few sections we exploit this fact by defining a simpler allocation object and a simpler process that mimics the pseudograph process we wish to analyse.

### 4.4 Transition probabilities

In the following, we discuss how, as a consequence of Lemma 4.3, \(\{M_t\}_{t \geq 0}\) can be modelled as a Markov chain where the states are restricted spaces of pseudographs. One way of representing this Markov chain, which perhaps clarifies what we mean by it, is as a directed graph, \(G_C\) (Definition 4.7): The vertices in \(G_C\) are the states of the chain, and each edge has a weight given by the transition probability between the two states. The transition probabilities are defined with respect to \textbf{EdgeDel}(\(k\)). Points in the process correspond to paths in \(G_C\).

There are two lemmas in this section. In Lemma 4.8 we give the transition probability with respect to \textbf{EdgeDel}(\(k\)) between each pair of restricted spaces, or vertices, in \(G_C\). We show that the transition probability between any two states, say \(\mathcal{M}_k(d, \nu, s)\) and \(\mathcal{M}'_k(d', \nu', s')\), is a function of \(d, \nu, s, d', \nu'\) and \(s'\) only. This will enable us, in Section 4.5, to replace this Markov chain by an even simpler one, in which the states are, essentially, the vectors \((d, \nu, s)\). The second lemma, Lemma 4.11, gives the analogous transition probabilities with respect to \textbf{EdgeOri}(\(k\)) between states in a Markov chain modelling \(\{N_t\}_{t \geq 0}\).

To model \(\{N_t\}_{t \geq 0}\) as a Markov chain we first define restricted spaces of partially oriented pseudographs (Definition 4.9), which will be the states in the chain. The transition probabilities for this chain are with respect to \textbf{EdgeOri}(\(k\)), and in Lemma 4.11 we show that they are closely related to those with respect to \textbf{EdgeDel}(2\(k + 1\)).
4.4.1 Transition probabilities for the $k$-core process

Let $\Omega$ be the probability space $\{M_t\}_{t \geq 0}$ (so $\Omega = \Omega_{\text{Core}(k)}$). For each point, or sequence of pseudographs, $(m_1, m_2, \ldots)$, of $\Omega$ (we use small letters here to distinguish pseudographs from their random counterparts), there is a sequence of vectors $(r_t, s_t, d_t) \in L_k \geq 0$ such that, for each $t$, $m_t$ has low degree sequence $d_t$, and $n - r_t$ heavy vertices containing $s_t$ points. By Lemma 4.3, as already observed, restricting to $D_{d_t}(n - r_t, s_t)$, $M_t$ has the distribution of $M_t = M_k(d_t, n - r_t, s_t)$, for each $t \geq 0$. Hence, for each element of $\Omega$, there is a corresponding sequence of restricted spaces $\{M_t\}_{t \geq 0}$.

On the other hand, for each restricted space $M_t = M_k(d_t, n - r_t, s_t)$ there is a weight function corresponding to the probability that $M_t$ has the distribution of $M_i$,

$$p_{M_i}(t) := \mathbb{P}(M_t \in M_i \mid M_0 \in M_0).$$

Let restricted spaces of the form $M_k(0, \nu, s)$, for any $\nu$ and $s$, be called terminal spaces, since if $M_t \in M_k(0, \nu, s)$ for some $\nu$ and $s$, then for all $T > t$, $M_T = M_t$ and so $M_T \in M_k(0, \nu, s)$.

Note that exactly 2 points are deleted in each application of $\text{EdgeDel}(k)$, unless $M_t$ is in a terminal space, so for $d(t) = (d_1, \ldots, d_r)$ and $T_i = \frac{1}{2}(2m - s_i - \sum_j d_j)$,

$$p_{M_i}(t) = \begin{cases} 
0 & \text{if } t < T_i \\
0 & \text{if } t > T_i \text{ and } d(t) \neq 0 \\
p_{M_i}(T_i) & \text{if } t > T_i \text{ and } d(t) = 0.
\end{cases}$$

So we may define $p_{M_i} := p_{M_i}(T_i)$.

These weights are determined by transition probabilities between pairs of restricted spaces.

**Definition 4.6** For probability spaces of pseudographs $M_1$ and $M_2$ define the transition probability, $T_{\text{ED}(k)}(M_1, M_2)$, with respect to $\text{EdgeDel}(k)$ to be

$$T_{\text{ED}(k)}(M_1, M_2) := \mathbb{P}(M' \in M_2 \mid M \in M_1),$$

where $M'$ is the result of applying $\text{EdgeDel}(k)$ to $M$. 57
Now, $\Omega_c$ can be represented as a weighted, directed graph.

**Definition 4.7** Define $G_C$, the graph of $\Omega_C$, to be the weighted, directed graph with a vertex for each restricted space, $M_k(d, \nu, s)$, and each directed edge, $(M_1, M_2)$, having weight $T_{\text{ED}(k)}(M_1, M_2)$. Edges with weight zero are omitted.

Hence, for each element of $\Omega_C$ there is a corresponding sequence of restricted spaces, $\{M_t\}_{t \geq 0}$, and a directed path in $G_C$ from $M_0$ to a terminal space.

Further, for each directed path, $(M_0, M_1, \ldots, M_T)$ in $G_C$, there is an associated probability which is the product of the weights for each edge on the path,

$$\Pr(M_i \in M_i \text{ for each } 0 \leq i \leq T) = \prod_{i=0}^{T-1} T_{\text{ED}(k)}(M_i, M_{i+1}),$$

and, for each restricted space $M_T$, $p_{M_T}$ is given by the sum of probabilities for each path in $G_C$ from $M_0$ to $M_T$ (note that, as a consequence of $\text{Core}(k)$ deleting exactly one edge in each time step, all such paths will have the same length),

$$p_{M_T} = \sum_{\mathcal{P}(M_0, M_T)} \prod_{i=0}^{T-1} T_{\text{ED}(k)}(M_i, M_{i+1}), \quad (4.1)$$

where $\mathcal{P}(M_0, M_T)$ is the set of paths in $G_C$ from $M_0$ to $M_T$.

Let $M_c$ be the output of $\text{Core}(k)$. For any pseudograph $H$ with low degree sequence $0$, let $\nu_H$ and $s_H$ be the number of heavy vertices and points in heavy vertices respectively, in $H$, and let $M_H = M_k(0, \nu_H, s_H)$. Then each such $M_H$ is a terminal space and the distribution of $M_c$ is given by

$$\Pr(M_c = H \mid M_0 \in M_0) = \Pr(M_c = H \mid M_c \in M_H)\Pr(M_c \in M_H \mid M_0 \in M_0) = \Pr(M = H \mid M \in M_H)p_{M_H}. \quad (4.2)$$

Now, $\Pr(M = H \mid M \in M_H)$ is determined by the allocation model and is independent of the process, and $p_{M_H}$, which is exactly the probability that $M_c$ has $\nu_H$ heavy vertices containing a total of $s_H$ points, is entirely determined by the transition probabilities.
Figure 4.1: A path in $G_C$ with edge weights $T_j = T_{ED(k)}(M_{j-1}, M_j)$. Each vertex represents a restricted subspace $M_k(d, \nu, s)$ for some $d$, $\nu$ and $s$, and each path corresponds to an element of $\Omega_C$.

Next, we will show that each transition probability, $T_{ED(k)}(M, M')$, with $M = M_k(d, \nu, s)$, can be expressed as a function of $\nu$, $s$ and $d_i$, $i = 1, \ldots, n - \nu$ only. That is the edge weights, as well as the restricted space for each vertex of $G_C$, are determined by the vector $(r, s, d)$.

There are a finite number of cases for which $T_{ED(k)}(M, M')$ is non-zero and these are treated separately as three general cases. Let $e^{(j)}$ denote the vector (of appropriate dimension for the context) with $j$th term 1 and zeros elsewhere.

**Lemma 4.8** For fixed $n$, $d = (d_1, \ldots, d_r) \in L_k^r$, $d' \in L_k^r$, and $s', s = \nu$ and $d_i$, $i = 1, \ldots, n - \nu$ only. Let $R_1 = \{i \in [r] : d_i > 0\}$ and let $r_1$ be the size of $R_1$.

For $i_1, i_2 \in R_1$ let $E_{1}^{(i_1, i_2)}$, $E_{2}^{(i_1)}$ and $E_{3}^{(i_1)}$ be the events

$$
E_{1}^{(i_1, i_2)} := \begin{cases} 
  r' = r, & s' = s \text{ and } d' = d - e^{(i_1)} - e^{(i_2)} \\
\end{cases}
$$

$$
E_{2}^{(i_1)} := \begin{cases} 
  r' = r, & s' = s - 1 \text{ and } d' = d - e^{(i_1)} \\
\end{cases}
$$

$$
E_{3}^{(i_1)} := \begin{cases} 
  r' = r + 1, & s' = s - k \text{ and } d'_i = d_i - e^{(i_1)}_i, \text{ } i = 1, \ldots, r, \text{ } d'_{r+1} = k - 1. \\
\end{cases}
$$
Let $E_0$ be the event

$$E_0 := [d' = d = 0 \text{ and } s' = s],$$

and let $E_4$ be the complement of the union of $E_0$, $E_1^{(i_1,i_2)}$, $E_2^{(i_1)}$, $E_3^{(i_1)}$ and all analogously defined events for each pair $(i_1, i_2) \in R_1 \times R_1$:

$$E_4 := \neg \bigcup_{i_1,i_2 \in R_1} (E_1^{(i_1,i_2)} \cup E_2^{(i_1)} \cup E_3^{(i_1)} \cup E_0).$$

Then, restricting to $E_0$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = 1. \quad (4.3)$$

Restricting to $E_1^{(i_1,i_2)}$, if $i_1 \neq i_2$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = \frac{d_{i_2} + d_{i_1}}{r_1(s - 1 + \sum_i d_i)}, \quad (4.4)$$

and if $i_1 = i_2$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = \frac{d_{i_2} - 1}{r_1(s - 1 + \sum_i d_i)}. \quad (4.5)$$

Restricting to $E_2^{(i_1)}$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = \frac{s - (n - r)km_{k,n-r,s,k}}{r_1(s - 1 + \sum_i d_i)}. \quad (4.6)$$

Restricting to $E_3^{(i_1)}$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = \frac{(n - r)km_{k,n-r,s,k}}{r_1(s - 1 + \sum_i d_i)}. \quad (4.7)$$

and restricting to $E_4$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = 0. \quad (4.8)$$

Where $m_{k,n-r,s,k}$ is given by (3.15).

**Proof.** If $d = 0$ then $\mathcal{M}$ is a terminal space so, restricting to $E_0$, $\mathcal{M}' = \mathcal{M}$ and (4.3) is clearly true. Assume $d \neq 0$. Let $M \in \mathcal{M}$ and let $M'$ be the result of applying $\text{EdgeDel}(k)$ to $M$. Let $e_1$ denote the first point of $M$ chosen in $\text{EdgeDel}(k)$, and let $e_2$ denote its mate. Then $e_1$ is in a vertex chosen u.a.r. of degree less than $k$, that is, a vertex in $R_1$ and, as a corollary to Lemma 4.2, $e_2$ is uniformly distributed across all other points in $M$. 60
Hence, for each \( i_1, i_2 \in [n] \),
\[
    P(e_1 \text{ is in vertex } i_1) = \begin{cases} 
        1/r_1 & i_1 \in R_1 \\
        0 & \text{otherwise},
    \end{cases} 
\] (4.9)

\[
    P(e_2 \text{ is in vertex } i_2 \mid e_1 \text{ is in vertex } i_1) = \frac{d(i_2) - \delta_{i_1,i_2}}{s-1 + \sum_i d_i}. 
\] (4.10)

The vertex containing the point \( e_2 \), say \( v_2 \), may be a heavy vertex, and
\[
P(d(v_2) = k) = \sum_{i=r+1}^{n} P(v_2 = i \mid d(i) = k \mid M \in \mathcal{M}) 
\]
\[
= \sum_{i=r+1}^{n} P(v_2 = i \mid d(i) = k) P(d(i) = k \mid M \in \mathcal{M}) 
\]
\[
= \frac{(n-r)km_{k,n-r,s,k}}{(s-1 + \sum_i d_i)}, 
\] (4.11)

by (3.16) in Corollary 3.10, and
\[
P(d(v_2) > k) = P(d(v_2) \geq k) - P(d(v_2) = k) 
\]
\[
= \frac{s - (n-r)km_{k,n-r,s,k}}{(s-1 + \sum_i d_i)}. 
\] (4.12)

There are exactly three cases to consider: \( e_2 \) is in a vertex of degree less than \( k \); \( e_2 \) is in a vertex of degree \( k \); or \( e_2 \) is in a vertex of degree greater than \( k \).

For the first case, w.l.o.g. suppose \( e_1 \) is in vertex \( i_1 \in R_1 \) and \( e_2 \) is in vertex \( i_2 \in R_1 \) (the following applies to any choice of \( i_1 \) and \( i_2 \)). Then \( M' \) has low degree sequence \( d^{(1)} := d - e^{(i_1)} - e^{(i_2)} \) and has \( n-r \) heavy vertices containing a total of \( s \) points. Hence, by Lemma 4.3, \( M' \) has the distribution of \( \mathcal{M}^{(1)} := \mathcal{M}_k(d^{(1)}, n-r, s) \). \( M' \) will also have the distribution of \( \mathcal{M}^{(1)} \) if \( e_i \) is in \( i_2 \) and \( e_2 \) is in \( i_1 \). These correspond exactly to \( e^{(i_1,i_2)}_1 \).

So, restricting to \( e^{(i_1,i_2)}_1 \), if \( i_1 \neq i_2 \),
\[
T_{ED^{(1)}}(\mathcal{M}, M') = T_{ED^{(1)}}(\mathcal{M}, \mathcal{M}^{(1)}) 
\]
\[
= P(M' \in \mathcal{M}^{(1)} \mid M \in \mathcal{M}) 
\]
\[
= P(e_1 \text{ in } i_1 \land e_2 \text{ in } i_2) + P(e_1 \text{ in } i_2 \land e_2 \text{ in } i_1) 
\]
\[
= \frac{d_{i_2}}{r_1(s-1 + \sum_i d_i)} + \frac{d_{i_1}}{r_1(s-1 + \sum_i d_i)}. 
\]
and if $i_1 = i_2$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = P(e_1 \text{ and } e_2 \text{ in } i_1) = \frac{d_{i_1} - 1}{r_1(s - 1 + \sum_i d_i)},$$

by (4.9) and (4.10), as required.

For the second case, w.l.o.g. suppose $e_1$ is in vertex $i_1$ and $e_2$ is in vertex $v_2$ with $d(v_2) > k$ (the following applies to any choice of $i_1$ and $v_2$). Then $M'$ has low degree sequence $d^{(2)} := d - e^{(i_1)}$ and has $n - r$ heavy vertices containing a total of $s - 1$ points. Hence, by Lemma 4.3, $M'$ has the distribution of $\mathcal{M}^{(2)} := \mathcal{M}_k(d^{(2)}, n - r, s - 1)$. This corresponds exactly to $\mathcal{E}_2^{(i_1)}$. So, restricting to $\mathcal{E}_2^{(i_1)}$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = T_{ED(k)}(\mathcal{M}, \mathcal{M}^{(2)}) = P(M' \in \mathcal{M}^{(2)} | M \in \mathcal{M}) = P(e_1 \text{ in } i_1 \land d(v_2) > k | M \in \mathcal{M}) = \frac{s - (n - r)km_{k,n-r,s,k}}{r_1(s - 1 + \sum_i d_i)},$$

by (4.9) and (4.12), as required.

For the final case, w.l.o.g. suppose $e_1$ is in vertex $i_1$ and $e_2$ is in vertex $v_2$ with $d(v_2) = k$. Then $M'$ has low degree sequence $d^{(3)} := (d^{(2)}_1, d^{(2)}_2, \ldots, d^{(2)}_r, k - 1)$ and has $n - r - 1$ heavy vertices containing a total of $s - k$ points. Hence, by Lemma 4.3, $M'$ has the distribution of $\mathcal{M}^{(3)} := \mathcal{M}_k(d^{(3)}, n - r - 1, s - k)$. This corresponds exactly to $\mathcal{E}_3^{(i_1)}$. So, restricting to $\mathcal{E}_3^{(i_1)}$,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = T_{ED(k)}(\mathcal{M}, \mathcal{M}^{(3)}) = P(M' \in \mathcal{M}^{(3)} | M \in \mathcal{M}) = P(e_1 \text{ in } i_1 \land d(v_2) = k | M \in \mathcal{M}) = \frac{(n - r)km_{k,n-r,s,k}}{r_1(s - 1 + \sum_i d_i)},$$

by (4.9) and (4.11), as required.
These are the only three cases when $d \neq 0$, so if $\mathcal{M}'$ is not analogous to $\mathcal{M}^{(1)}$, $\mathcal{M}^{(2)}$ or $\mathcal{M}^{(3)}$ for some $i_1$ and $i_2$, then this corresponds to event $\mathcal{E}_4$ and $T_{ED(k)}(\mathcal{M}, \mathcal{M}') = 0$ as required.

Note that, in summary of the arguments above,

$$T_{ED(k)}(\mathcal{M}, \mathcal{M}') = \sum_{i_1, i_2 \in R_1} I_{\xi(i_1, i_2)} P(e_1 \text{ in } i_1 \wedge e_2 \text{ in } i_2 )$$

$$+ \sum_{i_1 \in R_1} \left[ I_{\xi_2(i_1)} P(e_1 \text{ in } i_1 \wedge d(v_2) > k) \right]$$

$$+ I_{\xi_3(i_1)} P(e_1 \text{ in } i_1 \wedge d(v_2) = k),$$

(4.13)

where each probability is conditioned on $M \in \mathcal{M}$, $e_1$ and $e_2$ refer to the first and second points chosen in an application of $\text{EdgeDel}(k)$ to $M$ and $v_2$ refers to the vertex containing $e_2$.

### 4.4.2 Transition probabilities for the load balancing process

We now define probability spaces of partially oriented pseudographs so that the probability space determined by $\text{EdgeOri}(k)$ can be described.

**Definition 4.9** Let $\hat{\mathcal{M}}_l(d, j, \nu, s)$ be the probability space of partially oriented pseudographs with low load-degree sequence $(d, j)$, and with distribution induced by $\mathcal{M}_l(d - 2j, \nu, s)$, that is,

$$P(N | N \in \hat{\mathcal{M}}_l(d, j, \nu, s)) = P(\text{Unor}(N) | \text{Unor}(N) \in \mathcal{M}_l(d - 2j, \nu, s)).$$

In particular, this means that the elements of $\hat{\mathcal{M}}_l(d, j, \nu, s)$ are really equivalence classes of partially oriented pseudographs where $N \equiv H$ if $N$ and $H$ have the same low load-degree sequence and $\text{Unor}(N) = \text{Unor}(H)$.

Note that, by this definition, the location of out-points does not affect the distribution of partially oriented pseudographs. Out-points are, for the most part, ignored.
As described in Section 4.3, for some fixed \( r_0, d(0) \in L_{2k+1}^r \) and \( s_0 \) let \( N_0 \) be a pseudograph with the distribution of \( M_{2k+1}^r(d(0), n-r_0, s_0) \), and let \( N_t \) be the partially oriented pseudograph after \( \text{EdgeOri}(k) \) has been applied \( t \) times to \( N_0 \). Assume that \( N_{t+1} = N_t \) if \( \text{EdgeOri}(k) \) can not be applied to \( N_t \), that is if \( N_t \) has low load-degree sequence \( (d, j) \) with \( d_i - 2j_i = 0 \) or \( d_i = 2k + 1 \) for each \( i \).

Let

\[
\hat{E}_k := \bigcup_{r \geq 1} \{ (d, j) \in \hat{L}_{2k+1}^r : d_i - 2j_i = 0 \lor d_i = 2k + 1 \text{ for each } i = 1, \ldots, r \}.
\]

Then \( \text{EdgeOri}(k) \) cannot be applied to \( N_t \) if \( N_t \) has low load-degree sequence in \( \hat{E}_k \).

Let \( \Omega_L \) be the probability space \( \{ N_t \}_{t \geq 0} \). For each point, or sequence of partially oriented pseudographs, \( (n_1, n_2, \ldots) \) in \( \Omega_L \), there is a sequence \( \{ r_t, s_t, (d(t), j(t)) \in \hat{L}_{2k+1}^r \}_{t \geq 0} \) such that for each \( t \), \( n_t \) is in \( N_t = \hat{M}_{2k+1}^r(d(t), j(t), n-r_t, s_t) \). Hence, for each point in \( \Omega_L \), there is a corresponding sequence of restricted spaces \( \{ N_t \}_{t \geq 0} \). Further, by Corollary 4.5, restricting to \( \hat{D}_{d(t), j(t), n-r_t, s_t}, N_t \) has the distribution of \( N_t \).

Let restricted spaces \( \hat{M}_{2k+1}^r(d, j, \nu, s) \) with \( (d, j) \in \hat{E}_k \) be called terminal spaces, since if \( N_t \) is in a terminal space, \( N_t \), then for all \( T > t \), \( N_T = N_t \) and \( N_T \in N_t \).

In the same way as for \( \Omega_C \), \( \Omega_L \) can be represented as a weighted, directed graph, \( G_L \), with a vertex for each restricted space \( \hat{M}_{2k+1}^r(d, j, \nu, s) \), and each edge having a weight determined by a transition probability. Then, for each point in \( \Omega_L \), the corresponding sequence of restricted spaces, \( \{ N_t \}_{t \geq 0} \), is represented by a directed path from \( N_0 := \hat{M}_{2k+1}^r(d(0), 0, n-r_0, s_0) \) to a terminal space in \( G_L \).

**Definition 4.10** For probability spaces of partially oriented pseudographs \( N_1 \) and \( N_2 \) define the transition probability, \( T_{\text{EO}(k)}(N_1, N_2) \), with respect to \( \text{EdgeOri}(k) \) to be

\[
T_{\text{EO}(k)}(N_1, N_2) := P(N' \in N_2 \mid N \in N_1),
\]

where \( N' \) is the result of applying \( \text{EdgeOri}(k) \) to \( N \).
We now define four events similar to $E_0$, $E^{(i_1,i_2)}$, $E^{(i_1)}_2$ and $E^{(i_1)}_3$. For $(d,j) \in \hat{L}_{2k+1}$, $(d',j') \in \hat{L}'_{2k+1}$, $s$ and $s'$ let $R_1$ be the set of vertices containing greater than zero and less than $2k + 1$ free points and let $P_1$ be the set of vertices of highest priority with respect to $k$, as determined by $(d,j)$. That is,

$$R_1 = \{ i \in [r] : 0 < d_i - 2j_i < 2k + 1 \}$$

and

$$P_1 = \{ i \in [r] : Pri_k(d_i,j_i) = \min_{h \in [r]}(Pri_k(d_h,j_h)) \}.$$  

Let $r_1$ and $p_1$ be the size of $R_1$ and $P_1$ respectively. For $i_1 \in P_1$ and $i_2 \in R_1$ let $\hat{E}^{(i_1,i_2)}_1$, $\hat{E}^{(i_1)}_2$ and $\hat{E}^{(i_1)}_3$ be the following events:

$$\hat{E}^{(i_1,i_2)}_1 := [r' = r, s' = s \land d' = d + e^{(i_1)} - e^{(i_2)}, j' = j + e^{(i_1)}],$$
$$\hat{E}^{(i_1)}_2 := [r' = r, s' = s - 1 \land d' = d + e^{(i_1)}, j' = j + e^{(i_1)}],$$
$$\hat{E}^{(i_1)}_3 := [r' = r + 1, s' = s - 2k - 1 \land d'_i = d_i + e^{(i_1)}_i, i \in [r], d'_r + 1 = 2k,$$
$$j'_i = j_i + e^{(i_1)}_i, i \in [r], j'_{r+1} = 0].$$

Put simply, if $e_1$ is the point chosen in $\textbf{EdgeOri}(k)$ to become an in-point and $e_2$ is its mate then $\hat{E}^{(i_1,i_2)}_1$, $\hat{E}^{(i_1)}_2$ and $\hat{E}^{(i_1)}_3$ correspond respectively to the events that $e_2$ is in a vertex $i < r$, $e_2$ is in a heavy vertex of load-degree $2k + 1$, or in a heavy vertex of load-degree greater than $2k + 1$.

Let $\hat{E}$ be the union of these three events over all $i_1 \in P_1$ and $i_2 \in R_1$:

$$\hat{E} := \bigcup_{i_1 \in P_1, i_2 \in R_1} (\hat{E}^{(i_1,i_2)}_1 \cup \hat{E}^{(i_1)}_2 \cup \hat{E}^{(i_1)}_3).$$

Finally, let $\hat{E}_0$ be the event

$$\hat{E}_0 := [r' = r, s' = s, d' = d, j' = j \land (d,j) \in \hat{E}_k].$$

This corresponds to the event that $\textbf{EdgeOri}(k)$ cannot be performed.
Lemma 4.11 For fixed $n$, $(d,j) \in \hat{L}_{2k+1}$, $(d',j') \in \hat{L}'_{2k+1}$, $s$ and $s'$, let $N = \hat{M}_{2k+1}(d,j,n-r,s)$, $N' = \hat{M}_{2k+1}(d',j',n-r',s')$, $M = M_{2k+1}(d-2j,n-r,s)$ and $M' = M_{2k+1}(d'-2j',n-r',s')$. Then, if $(d,j) \in \hat{E}_k$,

$$T_{EO(k)}(N,N') = I_{\hat{E}_0},$$

and, if $(d,j) \notin \hat{E}_k$,

$$T_{EO(k)}(N,N') = \frac{r_1}{p_1} T_{ED(2k+1)}(M,M') I_{\hat{E}}.$$

Proof. If $(d,j) \in \hat{E}_k$ then $N$ is a terminal space and so $T_{EO(k)}(N,N') = I_{N_N} = I_{N'} = I_{\hat{E}_0}$, as required. Assume $(d,j) \notin \hat{E}_k$.

Let $N \in \mathcal{N}$ and let $N'$ be the result of applying $\text{EdgeOri}(k)$ to $N$. Now, $\text{Unor}(N) \in \mathcal{M}$ and if $N' \in N'$ then $\text{Unor}(N') \in \mathcal{M}'$. Let $M'$ be the result of applying $\text{EdgeDel}(2k+1)$ to $\text{Unor}(N)$ and let $e_1$ denote the first point of $\text{Unor}(N)$ chosen in $\text{EdgeDel}(2k+1)$, and let $e_2$ denote its mate. Similarly, let $f_1$ denote the first point of $N$ chosen in $\text{EdgeOri}(k)$, and let $f_2$ denote its mate. Then $f_1$ is in a vertex in $P_1$ chosen u.a.r. and, as a corollary to Lemma 4.2, since all possible labellings of the free points in $N$ determine a partially oriented pseudograph with the distribution of $N$, $f_2$ is uniformly distributed across all other free points in $N$. Hence, for each vertex $i_1,i_2 \in [n]$,

$$P(f_1 \text{ is in vertex } i_1) = \begin{cases} 1/p_1 & i_1 \in P_1 \\ 0 & \text{otherwise} \end{cases}$$

by (4.9), and

$$P(f_2 \text{ is in vertex } i_2 \mid f_1 \text{ is in } i_1) = \frac{d_L(i_2) - 2j_2 - \delta_{i_1,i_2}}{s - 1 + \sum d_i} = \frac{d(i_2) - \delta_{i_1,i_2}}{s - 1 + \sum d_i} = P(e_2 \text{ is in vertex } i_2 \mid e_1 \text{ is in } i_1),$$

by (4.10), where $d(i_2)$ refers to the degree of $i_2$ in $\text{Unor}(N)$. 

66
Hence, for $i_1 \in P_1$,

$$
P(f_1 \text{ in } i_1 \land f_2 \text{ in } i_2) = P(f_1 \text{ in } i_1)P(f_2 \text{ in } i_2 \mid f_1 \text{ in } i_1)
= \frac{r_1}{p_1}P(e_1 \text{ in } i_1)P(e_2 \text{ in } i_2 \mid e_1 \text{ in } i_1)
= \frac{r_1}{p_1}P(e_1 \text{ in } i_1 \land e_2 \text{ in } i_2). \quad (4.14)
$$

Let $u_2$ be the vertex containing $f_2$. If $u_2 > r$ then $u_2$ is a heavy vertex with $d^L(u_2) \geq 2k+1$ and $d^-(u_2) = 0$, so $d(u_2) \geq 2k+1$ in $Unor(N)$ also. Conversely, if $d(u_2) \geq 2k+1$ in $Unor(N)$, then $u_2$ is a heavy vertex in $N$ and we have

$$
P(f_1 \text{ in } i_1 \land u_2 > r \text{ with } d^L(u_2) > 2k+1) = P(f_1 \text{ in } i_1)P(u_2 > r, d^L(u_2) > 2k+1)
= \frac{r_1}{p_1}P(e_1 \text{ in } i_1)P(d(v_2) > 2k+1)
= \frac{r_1}{p_1}P(e_1 \text{ in } i_1 \land d(v_2) > 2k+1), \quad (4.15)
$$

and similarly,

$$
P(f_1 \text{ in } i_1 \land u_2 > r, d^L(u_2) = 2k+1) = \frac{r_1}{p_1}P(e_1 \text{ in } i_1 \land d(v_2) = 2k+1). \quad (4.16)
$$

Let $f = d - 2j$ and $f' = d' - 2j'$ and let $\mathcal{E}_1^{(i_1,i_2)}, \mathcal{E}_2^{(i_1)}$ and $\mathcal{E}_2^{(i_1)}$ be the events defined in Lemma 4.8 for $f$, $f'$, $s$ and $s'$. Then

$$
\mathcal{E}_1^{(i_1,i_2)} \subseteq [r' = r, s' = s \text{ and } d' - 2j' = d - 2j - e^{(i_1)} - e^{(i_2)}]
= [r' = r, s' = s \text{ and } f' = f - e^{(i_1)} - e^{(i_2)}]
= \mathcal{E}_1^{(i_1,i_2)}.
$$

So $\mathcal{E}_1^{(i_1,i_2)} \subseteq \mathcal{E}_1^{(i_1,i_2)} \cap \mathcal{E}$ since $\mathcal{E}_1^{(i_1,i_2)} \subseteq \mathcal{E}$ by definition. Conversely,

$$
\mathcal{E}_1^{(i_1,i_2)} \cap \mathcal{E} = \mathcal{E}_1^{(i_1,i_2)} \cap \bigcup_{h_1 \in P_1, h_2 \in R_1} (\mathcal{E}_1^{(h_1,h_2)} \cup \mathcal{E}_2^{(h_1)} \cup \mathcal{E}_3^{(h_1)})
= \mathcal{E}_1^{(i_1,i_2)} \cap (\mathcal{E}_1^{(i_1,i_2)} \cup \mathcal{E}_1^{(i_2,i_1)})
= \mathcal{E}_1^{(i_1,i_2)} \cap \mathcal{E}_1^{(i_1,i_2)} \subseteq \mathcal{E}_1^{(i_1,i_2)}.
$$

67
Hence

\[ \hat{E}^{(i_1, i_2)}_1 = E^{(i_1, i_2)}_1 \cap \hat{E} . \]

Similarly, \( \hat{E}^{(i_1)}_2 = E^{(i_1)}_2 \cap \hat{E} \) and \( \hat{E}^{(i_1)}_3 = E^{(i_1)}_3 \cap \hat{E} \), and so

\[ I^{(i_1, i_2)}_E = I^{(i_1, i_2)}_E \cap \hat{E} \]
\[ I^{(i_1)}_E = I^{(i_1)}_E \]
\[ I^{(i_1)}_E = I^{(i_1)}_E \]

Hence, combining (4.14), (4.15) and (4.16),

\[
\begin{align*}
T_{\text{EO}(k)}(N, N') &= \sum_{i_1 \in P_1, i_2 \in R_1} I_{E^{(i_1, i_2)}_1} P(f_1 \text{ in } i_1, f_2 \text{ in } i_2) \\
&+ \sum_{i_1 \in P_1} I_{E^{(i_1)}_2} P(f_1 \text{ in } i_1 \text{ and } u_2 > r \text{ with } d^L(u_2) > 2k + 1) \\
&+ \sum_{i_1 \in P_1} I_{E^{(i_1)}_3} P(f_1 \text{ in } i_1 \text{ and } u_2 > r \text{ with } d^L(u_2) = 2k + 1) \\
&= \sum_{i_1 \in P_1, i_2 \in R_1} I_{E^{(i_1, i_2)}_1} I_{E^{(i_1, i_2)}_1} P(e_1 \text{ in } i_1 \text{ and } e_2 \text{ in } i_2) \\
&+ \sum_{i_1 \in P_1} I_{E^{(i_1)}_2} I_{E^{(i_1)}_2} P(e_1 \text{ in } i_1 \text{ and } d(v_2) > 2k + 1) \\
&+ \sum_{i_1 \in P_1} I_{E^{(i_1)}_3} I_{E^{(i_1)}_3} P(e_1 \text{ in } i_1 \text{ and } d(v_2) = 2k + 1) \\
&= I_{E^{(i_1)}_1} P_1 T_{\text{ED}(2k + 1)}(M, M'),
\end{align*}
\]

by (4.13), which concludes the proof. \( \square \)

### 4.5 Partial pre-allocations

We have shown in Lemma 4.2 that the pairing of points in a pseudograph with the distribution of a restricted space is uniformly distributed on all possible pairings. We have also shown, in Lemmas 4.8 and 4.11, that not only is each vertex of \( G_C \) and \( G_L \) determined by the vector \((d, \nu, s)\) or \((d, j, \nu, s)\) respectively, but the edge weights are also determined by these vectors. As a consequence, \( \Omega_C \) can be modelled as a Markov chain on the vector \((d, \nu, s)\), and \( \Omega_L \) can be modelled as a Markov chain on the vector \((d, j, \nu, s)\).
In other words, we may model $\Omega_C$ and $\Omega_L$ as processes on pseudographs which, apart from the information specified by $(d, \nu, s)$ or $(d, j, \nu, s)$, are kept random. The random parts of the pseudograph can be gradually revealed, in each step of the process, as required, and according to the expected distributions.

In this section we define allocation-like representations of the vectors, called partial pre-allocations, that correspond to these partly fixed, partly random pseudographs: The low degree sequence is given, as is the number of heavy vertices and the sum of degrees of heavy vertices, but the pairing of points is not specified, nor is the degree sequence of heavy vertices. Each partial pre-allocation is exactly determined by a vector of the form $(d, \nu, s)$ or $(d, j, \nu, s)$, and corresponds uniquely to a restricted space of pseudographs, $\mathcal{M}_k(d, \nu, s)$ or $\mathcal{M}_{2k+1}(d, j, \nu, s)$.

We also define the edge generation and deletion or orientation steps, $\text{GenDel}(k)$ and $\text{GenOri}(k)$, for partial pre-allocations, by which our new processes shall be determined. In each of these steps a pairing of two points is generated, the resulting edge is deleted or oriented, and with a certain probability some more of the allocation, to low degree vertices only, is generated. In Section 4.6, we shall define two algorithms, $\text{ACore}(k)$ and $\text{ALoad}(k)$, based around $\text{GenDel}(k)$ and $\text{GenOri}(k)$, and the two processes, $\Omega_{\text{ACore}(k)}$ and $\Omega_{\text{ALoad}(k)}$, which they determine.

To prove our claim that the $\Omega_{\text{ACore}(k)}$ and $\Omega_{\text{ALoad}(k)}$ model $\Omega_C$ and $\Omega_L$, and hence can be analysed to reveal the distribution of the output of $\text{Core}(k)$ and $\text{Load}(k)$ (Lemmas 4.23 and 4.25), we consider the transition probabilities with respect to $\text{GenDel}(k)$ and $\text{GenOri}(k)$. In Lemma 4.17 we show that the transition probability with respect to $\text{GenDel}(k)$ between a pair of partial pre-allocations, is the same as the transition probability with respect to $\text{EdgeDel}(k)$, between the corresponding pair of restricted spaces. In Lemma 4.19 we show the equivalent result for $\text{GenOri}(k)$ and $\text{EdgeOri}(k)$.

**Definition 4.12** The pre-allocation, $A(d)$, with degree sequence $d = (d_1, \ldots, d_n)$ is a set of $n$ vertices with each vertex, $i = 1, \ldots, n$ containing $d_i$ unlabelled points.
Recall that an allocation is an assignment of a set of labelled points, \([2m]\) into a set of
labelled cells, \([n]\) that we shall call vertices. A pseudograph on vertex set \([n]\) is obtained
when the pairs of points in the canonical pairing, \(C := \{(2i - 1, 2i); i \in [m]\}\), are identified
with edges.

An allocation can be generated from a pre-allocation, \(A\), by labelling all the points in
\(A\). Because our concern is only with the corresponding pseudograph, in which the points
are unlabelled, it is not necessary to explicitly label each of the points, only to generate
a pairing of the points. By Lemma 4.2, any uniformly random pairing of the points in
an allocation determines a pseudograph with the distribution of \(M(n, m)\) restricted to
pseudographs with degree sequence \(d\). Hence a pseudograph with this distribution can be
generated from \(A(d)\) by repeatedly choosing a point \(e\), then choosing a second point u.a.r.
to be the mate of \(e\), until all points are paired.

In each of \(\text{GenDel}(k)\) and \(\text{GenOri}(k)\), one pair of the pairing will be generated and
then, respectively, deleted or oriented. For a vertex \(i\) in a pre-allocation, this is described
as follows:

**Expose a point** \(e_1\) **in** \(i\).

**Expose the mate** \(e_2\) **of** \(e_1\).

Here, **expose a point** \(e_1\) **in** \(i\) simply means “choose an unpaired point in \(i\)” , whilst **expose the mate** \(e_2\) **of** \(e_1\) means “choose a second unpaired point u.a.r. to be the mate of the first.”

Clearly, repetition of these steps will generate a uniformly random pairing of the points.

In terms of generating an allocation from a pre-allocation, **expose a point** \(e_1\) **in** \(i\) can be
interpreted as “allocate a label chosen u.a.r. to an unlabelled point in \(i\)” , whilst **expose the mate** \(e_2\) **of** \(e_1\) can be interpreted as “allocate the label corresponding to the mate of \(e_1\) in
the canonical pairing, to an as yet unlabelled point, chosen u.a.r.” Then each time a point
is labelled, either the label or the point is chosen u.a.r. and consequently, the pairing of
the points is generated u.a.r.

The following definition of the partially oriented pre-allocation is independent of the lo-
cations of out-points, and labels of in-points are not explicitly given. A partially oriented pre-allocation is determined by the load-degree sequence, on which out-points have no effect. Out-points also have no bearing on \textbf{EdgeOri}(k), nor shall they have on \textbf{GenOri}(k). Further, we have defined the distribution of partially oriented pseudographs to be independent of the locations of out-points (Definition 4.9). All that matters is the location of in-points and that in-points and out-points have already been paired.

**Definition 4.13** The (partially oriented) pre-allocation, \( A(d,j) \), with load-degree sequence \((d,j) = (d_1,\ldots,d_n,j_1,\ldots,j_n) \) is a set of \( n \) vertices with each vertex, \( i = 1,\ldots,n \) containing \( d_i - 2j_i \) unlabelled points and \( j_i \) in-points. Each in-point has a mate, which is an out-point and may be in any vertex (two partially oriented pre-allocations are equivalent provided they have the same load-degree sequence).

Now, to correspond to the vectors \((d,\nu,s)\), we define partial pre-allocations. These are pre-allocations which only contain points in low degree vertices, and in a special vertex, \( n + 1 \). Each partial pre-allocation represents a pseudograph for which the pairing of points is random, the low degree sequence is fixed, as are the numbers of heavy vertices and points in heavy vertices, but the degree sequence of heavy vertices is random. The points in vertex \( n + 1 \) represent the points in heavy vertices of the pseudograph. These points “wait” in vertex \( n + 1 \) until their allocation is generated, according to the edge processing step we define next.

**Definition 4.14** For fixed \( n \) and \( k \), positive integers \( \nu \) and \( s > \nu k \), and \( d = (d_1,\ldots,d_{n-\nu}) \in \mathbb{L}_k^{n-\nu} \), the partial pre-allocation, \( B(d,\nu,s) \) is the pre-allocation \( A(\bar{d}) \), where \( \bar{d} \) is the \( n + 1 \) term vector \( \bar{d} = (d_1,\ldots,d_{n-\nu},0,\ldots,0,s) \).

The partial pre-allocation \( B(d,\nu,s) \) is associated with the restricted space \( \mathcal{M}_k(d,\nu,s) \) of \( \Omega_c \) in a natural and obvious way.
Definition 4.15 For fixed $n$ and $k$, positive integers $\nu$ and $s > \nu(2k + 1)$, and $(d, j) \in \hat{L}_{2k+1}^{n-\nu}$, the (partially oriented) partial pre-allocation, $\hat{B}(d, j, \nu, s)$ is the pre-allocation $A(\bar{d}, \bar{j})$ where $(\bar{d}, \bar{j})$ is the 2n+2 term vector $(\bar{d}, \bar{j}) = (d_1, \ldots, d_{n-\nu}, 0, \ldots, 0, s, j_1, \ldots, j_{n-\nu}, 0, \ldots, 0)$.

The partial pre-allocation $\hat{B}(d, j, \nu, s)$ is associated with the restricted space $\hat{M}_{2k+1}(d, j, \nu, s)$ of $\Omega_k$.

Now we define the steps GenDel($k$) and GenOri($k$) for $B(d, \nu, s)$ and $\hat{B}(d, j, \nu, s)$, which will have the same transition probabilities as EdgeDel($k$) and EdgeOri($k$) for $M_k(d, \nu, s)$ and $\hat{M}_{2k+1}(d, j, \nu, s)$. As well as generating a pairing of the points, these steps need to generate an allocation of the points in vertex $n + 1$ to vertices $n - \nu + 1$ to $n$.

In GenDel($k$), this will be done by moving points from vertex $n + 1$ to vertex $n - \nu + 1$ with the same probability that, in a pseudograph subject to EdgeDel($k$), a point in a vertex of degree $k$ is deleted, so that that vertex is no longer a heavy vertex. Similarly in GenOri($k$).

Let
\[ q_{\nu, s, l} := \frac{\nu m_{\nu, s, l}}{s}, \]
where $m_{\nu, s, l}$ is given by (3.15).

GenDel($k$) for $B = B(d, \nu, s)$ with $d \in L_k^{n-\nu}$.

Choose a vertex $i$ in $B$ with $0 < d(i) < k$, u.a.r.
Expose a point $e_1$ in $i$.

Expose the mate $e_2$ of $e_1$.

If $e_2$ is in vertex $n+1$, then with probability $q_{\nu,s,k}$,

Move $k-1$ unlabelled points from vertex $n+1$ into vertex $n-\nu+1$.

Delete $e_1$ and $e_2$.

$\text{GenOri}(k)$ for $B = \hat{B}(d,j,\nu,s)$ with $(d,j) \in \hat{L}_{2k+1}$.

Choose a vertex $i$ in $B$ of highest priority w.r.t. $k$, u.a.r.

Expose a free point $e_1$ in $i$.

Expose the mate $e_2$ of $e_1$.

If $e_2$ is in vertex $n+1$, then with probability $q_{\nu,s,2k+1}$,

Move $e_2$ and $2k$ unlabelled points from vertex $n+1$ into vertex $n-\nu+1$.

Make $e_1$ an in-point and $e_2$ an out-point.

Next we define the transition probabilities for these steps and show that they correspond exactly to those for $\text{EdgeDel}(k)$ and $\text{EdgeOri}(k)$.

**Definition 4.16** For partial pre-allocations $C$ and $C'$ define the transition probability, $T_{\text{GD}(k)}(C,C')$, with respect to $\text{GenDel}(k)$ to be

$$T_{\text{GD}(k)}(C,C') := P(C' = B' | C = B),$$

where $B'$ is the result of applying $\text{GenDel}(k)$ to $B$.

**Lemma 4.17** For fixed $n$, $d \in L_k^r$, $d' \in L_{k}^r$, $s$ and $s'$, let $\mathcal{M} = \mathcal{M}_k(d,n-r,s)$ and $\mathcal{M}' = \mathcal{M}_k(d',n-r',s')$, and let $B = B(d,n-r,s)$ and $B' = B(d',n-r',s')$. Then

$$T_{\text{Ed}(k)}(\mathcal{M},\mathcal{M}') = T_{\text{GD}(k)}(B,B').$$

**Proof.** If $d = 0$, then $T_{\text{Ed}(k)}(\mathcal{M},\mathcal{M}') = I_{\mathcal{E}_0}$, where $\mathcal{E}_0$ is the event that $d' = d = 0$ and $s' = s$. Clearly $T_{\text{GD}(k)}(B,B') = I_{\mathcal{E}_0}$ also, so the lemma holds. Assume that $d \neq 0$. 73
Let \( r_1, i_1, i_2, \mathcal{E}_1^{(i_1,i_2)}, \mathcal{E}_2^{(i_1)}, \mathcal{E}_3^{(i_1)}, \mathcal{E}_4, \mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \mathbf{d}^{(3)}, \mathcal{M}^{(1)}, \mathcal{M}^{(2)} \) and \( \mathcal{M}^{(3)} \) be as in Lemma 4.8. Let \( B^{(1)} = B(d^{(1)}, n-r, s) \), \( B^{(2)} = B(d^{(2)}, n-r, s-1) \) and \( B^{(3)} = B(d^{(3)}, n-r-1, s-k) \).

Suppose \( B \) is subjected to \( \text{GenDel}(k) \). Let \( f_1 \) be the first point chosen and \( f_2 \) its mate, and let \( R \) be the event that points are moved from vertex \( n+1 \) to vertex \( r+1 \).

There are exactly three cases: \( f_2 \) is in a vertex \( i \leq r \); \( f_2 \) is in \( n+1 \) and \( R \) does not occur; \( f_2 \) is in \( n+1 \) and \( R \) occurs.

For the first case suppose w.l.o.g. that \( f_1 \) is in \( i_1 \) and \( f_2 \) is in \( i_2 \), or visa versa. Then
\[
B' = B^{(1)}
\]
and this corresponds exactly to \( \mathcal{E}_1^{(i_1,i_2)} \). So, restricting to \( \mathcal{E}_1^{(i_1,i_2)} \), if \( i_1 \neq i_2 \),
\[
T_{\text{GD}(k)}(B, B^{(1)}) = \mathbb{P}(f_1 \text{ in } i_1 \land f_2 \text{ in } i_2) + \mathbb{P}(f_1 \text{ in } i_2 \land f_2 \text{ in } i_1) = \frac{d_{i_2}}{r_1(s-1+\sum_i d_i)} + \frac{d_{i_1}}{r_1(s-1+\sum_i d_i)} = T_{\text{ED}(k)}(\mathcal{M}, \mathcal{M}^{(1)}),
\]
by (4.4), as required. If \( i_1 = i_2 \),
\[
T_{\text{GD}(k)}(B, B^{(1)}) = \mathbb{P}(f_1 \text{ in } i_1 \land f_2 \text{ in } i_1) = \frac{d_{i_2} - 1}{r_1(s-1+\sum_i d_i)} = T_{\text{ED}(k)}(\mathcal{M}, \mathcal{M}^{(1)}),
\]
by (4.5), as required.

For the second case suppose w.l.o.g. that \( f_1 \) is in \( i_1 \), \( f_2 \) is in \( n+1 \), and not \( R \). Then \( B' = B^{(2)} \) and this corresponds exactly to \( \mathcal{E}_2^{(i_1)} \). So, restricting to \( \mathcal{E}_2^{(i_1)} \),
\[
T_{\text{GD}(k)}(B, B^{(2)}) = \mathbb{P}(f_1 \text{ in } i_1 \land f_2 \text{ in } n+1 \land \neg R) = \frac{s}{r_1(s-1+\sum_i d_i)}(1 - q_{(n-r),s,k})
\]
\[
= \frac{s - k(n-r)m_{k,n-r,s,k}}{r_1(s-1+\sum_i d_i)} = T_{\text{ED}(k)}(\mathcal{M}, \mathcal{M}^{(2)}),
\]
by (4.17) and (4.6), as required.
For the third case suppose w.l.o.g. that \( f_1 \) is in \( i_1 \), \( f_2 \) is in \( n + 1 \) and \( R \) occurs. Then \( B' = B^{(3)} \) and this corresponds exactly to \( \mathcal{E}_3^{(i_1)} \). So, restricting to \( \mathcal{E}_3^{(i_1)} \),

\[
T_{GD(k)}(B, B^{(3)}) = P(f_1 \text{ in } i_1 \land f_2 \text{ in } n + 1 \land R)
\]

\[
= \frac{s}{r_1(s - 1 + \sum_i d_i)} q(n-r),s,k
\]

\[
= \frac{k(n-r)m_{k,n-r,s,k}}{r_1(s - 1 + \sum_i d_i)}
\]

\[
= T_{ED(k)}(M, M^{(3)})
\]

by (4.17) and (4.7), as required.

These are the only three cases, so if \( B' \) is not analogous to \( B^{(1)} \), \( B^{(2)} \) or \( B^{(3)} \) for some \( i_1 \) and \( i_2 \), then this corresponds to event \( \mathcal{E}_4 \) and \( T_{GD(k)}(B, B') = 0 = T_{ED(k)}(M, M') \) as required.

**Definition 4.18** For partially oriented partial pre-allocations \( C \) and \( C' \) define the transition probability, \( T_{GO(k)}(C, C') \), with respect to \( \text{GenOri}(k) \) to be

\[
T_{GO(k)}(C, C') := P(C' = B' \mid C = B),
\]

where \( B' \) is the result of applying \( \text{GenOri}(k) \) to \( B \).

**Lemma 4.19** For fixed \( n, (d, j) \in \hat{L}_{2k+1}, (d', j') \in \hat{L}'_{2k+1}, s \) and \( s' \), let \( N = \hat{M}_{2k+1}(d, j, n-r, s) \) and \( N' = \hat{M}_{2k+1}(d', j', n-r', s') \), and let \( C = \hat{B}(d, j, n-r, s) \) and \( C' = \hat{B}(d', j', n-r', s') \). Then

\[
T_{EO(k)}(N, N') = T_{GO(k)}(C, C').
\]

The proof is omitted as it is straightforward using similar arguments as in the proof of Lemma 4.17.

**Corollary 4.20** For fixed \( n, (d, j) \in \hat{L}_{2k+1}, (d', j') \in \hat{L}'_{2k+1}, s \) and \( s' \), let \( C = \hat{B}(d, j, n-r, s) \), \( C' = \hat{B}(d', j', n-r', s') \) and let \( B = B(d - 2j, n-r, s) \) and \( B' = B(d' - 2j', n-r', s') \). Then, if \( (d, j) \in \hat{E}_k \),

\[
T_{GO(k)}(C, C') = I_{\hat{E}_0},
\]

75
and if \((d, j) \notin \hat{E}_k\),
\[
T_{\text{GO}(k)}(C, C') = \frac{r_1}{p_1} T_{\text{GD}(2k+1)}(B, B') I_{\hat{E}},
\]
where \(r_1, p_1, \hat{E} \) and \(\hat{E}_0\) are as in Lemma 4.11.

**Proof.** This is immediate from Lemmas 4.11, 4.17 and 4.19.

### 4.6 The partial pre-allocation processes

Two complete algorithms, \textbf{ACore}(k) and \textbf{ALoad}(k), which incorporate \textbf{GenDel}(k) and \textbf{GenOri}(k) respectively, are now defined. These determine the random processes \(\Omega_{\text{ACore}(k)}\) and \(\Omega_{\text{ALoad}(k)}\). Each algorithm starts with a particular partial pre-allocation, and ends at a finite time, \(T_s\). At this time, the process becomes constant. This constant state, \(B(T_s)\), is the final state of the process.

A *step of the process* refers to a step in the algorithm, that is, one iteration of the repeat loop, in which \(t\) is incremented by 1, and one pair of points is either deleted or oriented. *Step t* is the one in which time increments from \(t\) to \(t+1\). Say that a point \(p\) is *processed at time \(t\) if, in step \(t\), \(p\) is either deleted or becomes an in-point or an out-point. Say that a vertex, \(v\), is processed in step \(t\) if \(v\) is the vertex chosen to have one of its points exposed in step \(t\). Note that a vertex needs to be chosen as a low degree vertex to be processed.

#### 4.6.1 The \(k\)-core process

The process \(\Omega_{\text{ACore}(k)}\) is defined on partial pre-allocations \(B(d, n-r, s)\) with \(d \in L_k^r\), as per Definition 4.14. Recall that vertex \(n + 1\) in such a partial pre-allocation contains \(s\) points that represent all the points in the \(n - r\) heavy vertices of pseudographs with \(n\) vertices, \(\frac{1}{2}(s + \sum d_i)\) edges and low degree sequence \(d\). These \(s\) points are allocated dynamically to vertices \(r + 1\) to \(n\), unless they are deleted first.

Each step in the following is an application of \textbf{GenDel}(k), and \(q_{\nu, s, k}\) is given by (4.17).
Algorithm 4.21 ACore(k)

Fix $n$ and $m$.
Fix $d^{(0)} \in L^r_k$ with $r < n$ and $2m - \sum_i d_i^{(0)} > k(n - r)$.
Let $r(0) = r$ and $s(0) = 2m - \sum_i d_i^{(0)}$.
Let $B(0) = B(d^{(0)}, n - r(0), s(0))$.
Repeat the following:

Choose a vertex $i$ in $B(t)$ with $0 < d(i) < k$, u.a.r.
Expose a point $e_1$ in $i$.
Expose the mate $e_2$ of $e_1$.
If $e_2$ is in vertex $n + 1$, then with probability $q_{n-r(t),s(t),k}$,
Move $k - 1$ unlabelled points from vertex $n + 1$ into vertex $r(t) + 1$.
$s(t + 1) = s(t) - k$.
$r(t + 1) = r(t) + 1$.
Delete $e_1$ and $e_2$.
$t \leftarrow t + 1$.

Until all vertices $i \leq n$ of $B(t)$ have degree 0.
Output $B(t)$.
End.

ACore($k$) is well defined by the following.

Lemma 4.22 For all $t \geq 0$,
\[ s(t) \geq k(n - r(t)). \] (4.20)

Proof. By definition, $s(0) > k(n - r(0))$, so assume that for some $t$, (4.20) holds. If $e_2$ is not in vertex $n + 1$ then $s(t + 1) = s(t)$ and $r(t + 1) = r(t)$ so (4.20) still holds at time $t + 1$. If $e_2$ is in vertex $n + 1$ then either
\[ r(t + 1) = r(t) + 1 \quad \text{and} \quad s(t + 1) = s(t) - k \quad \text{or} \]
\[ r(t + 1) = r(t) \quad \text{and} \quad s(t + 1) = s(t) - 1. \]
If the former then
\[ s(t + 1) \geq k(n - r(t)) - k \]
\[ = k(n - r(t) - 1) \]
\[ = k(n - r(t + 1)). \]

If \( s(t) = k(n - r(t)) \) then
\[ q_{n-r(t), s(t), k} = \frac{k(n - r(t))m_{k,n-r(t), s(t), k}}{s(t)} \]
\[ = 1, \]
since \( m_{l,\nu,\nu, l} = 1 \), so the latter will not occur. If \( s(t) > k(n - r(t)) \) and the latter occurs, then
\[ s(t + 1) > k(n - r(t)) - 1 \]
\[ \geq k(n - r(t + 1)). \]

Hence in all cases \( s(t + 1) \geq k(n - r(t + 1)) \) and the result follows by induction on \( t \).

Let \( \Omega_{\text{ACore}(k)} := \{B(t)\}_{t \geq 0} \), and let \( G_{\text{AC}} \) be the graph of \( \Omega_{\text{ACore}(k)} \), in analogy with Definition 4.7 for \( G_{\text{C}} \). Define \( T_s \) to be the time when \( \text{ACore}(k) \) ends. Equivalently, \( T_s \) is the random time at which \( \Omega_{\text{ACore}(k)} \) becomes constant.

\[ T_s := \min\{t : \text{all vertices } i \leq n \text{ of } B(t) \text{ have degree 0}\}. \tag{4.21} \]

Let \( T_f := m, \) half the number of points in \( B(0) \). Two points are deleted in each step, so \( T_f \) is an upper bound for \( T_s \).

The following lemma summarises the correspondence between \( \Omega_{\text{ACore}(k)} \) and \( \Omega_{\text{C}} \). In particular, if \( s(T_s) = 0 \) this indicates that no \( k \)-core exists in a corresponding pseudograph. Otherwise, a \( k \)-core with \( n - r(T_s) \) vertices and average degree \( \frac{s(T_s)}{2(n-r(T_s))} \) exists.

**Lemma 4.23** Assume that \( B(0) \) is the partial pre-allocation associated with \( M \in \mathcal{M}(n,m) \). Then \( B(T_s) \) has the same distribution as the partial pre-allocation associated with the \( k \)-core of \( M \). In particular, the \( k \)-core of \( M \) has the distribution of \( \mathcal{M}_k(0, \nu, S) \), where \( \nu \) and \( S \) are random variables for \( n - r(T_s) \) and \( s(T_s) \) respectively.
Proof. Condition on \( M \in \mathcal{M}_0 = \mathcal{M}(\mathbf{d}^{(0)}, n - r_0, s_0) \) for some \( r_0 < n, \mathbf{d}^{(0)} \in L_k^{r_0} \) with \( s_0 = 2m - \sum_i d_i^{(0)} > k(n - r) \). Let \( B(0) = B(\mathbf{d}^{(0)}, n - r(0), s_0) \).

Let \( M_k \) be the \( k \)-core of \( M \). Then \( M_k \) has the same distribution as \( M_c \), the output of \( \text{Core}(k) \) applied to \( M_0 = M \). Moreover, as a corollary to Lemma 4.3, restricting \( M_c \) to \( D_{0,\nu,s} \), \( M_c \) has the distribution of \( \mathcal{M}_k(0, \nu, s) \).

Let \( B_k \) be the partial pre-allocation associated with \( M_k \). Then for any \( s, \nu \) and \( n - \nu \) term vector \( \mathbf{0} \), conditioning on \( M_0 \in \mathcal{M}_0 \),

\[
P(B_k = B(0, \nu, s)) = P(M_k \in \mathcal{M}_k(0, \nu, s)) = P(M_c \in \mathcal{M}_k(0, \nu, s)) = p_{\mathcal{M}_k(0, \nu, s)},
\]

by definition. Let \( \mathcal{P}(\mathcal{M}_0, \mathcal{M}) \) be the set of paths in \( G_C \) from \( \mathcal{M}_0 \) to \( \mathcal{M} \), and let \( \mathcal{P}(B_0, B) \) be the set of paths in \( G_{AC} \) from \( B(0) \) to \( B \). Now, by (4.1),

\[
p_{\mathcal{M}_k(0, \nu, s)} = \sum_{\mathcal{P}(\mathcal{M}_0, \mathcal{M})} T_f^{T_j-1} \prod_{i=0}^{T_j-1} T_{\text{ED}(k)}(\mathcal{M}_i, \mathcal{M}_{i+1})
\]

\[
= \sum_{\mathcal{P}(B_0, B)} \prod_{i=0}^{T_j-1} T_{\text{GD}(k)}(B_i, B_{i+1}),
\]

by Lemma 4.17 and, since \( G_C \) and \( G_{AC} \) have corresponding sets of vertices, for each path in \( G_C \) from \( \mathcal{M}_0 \) to \( \mathcal{M}_k(0, \nu, s) \) there is a corresponding path in \( G_{AC} \) from \( B(0) \) to \( B(0, \nu, s) \).

Finally, by writing an equation analogous to (4.1) for \( G_{AC} \), we easily see that the right hand side is equal to \( P(B(T_s) = B(0, \nu, s)) \). Hence

\[
P(B_k = B(0, \nu, s)) = P(B(T_s) = B(0, \nu, s)),
\]

and \( B_k \) has the distribution of \( B_{T_s} \). This is true conditioning on any appropriate \( \mathcal{M}_0 \), so holds for \( M \in \mathcal{M}(n, m) \).

The second statement of the lemma follows from the first and the corollary to Lemma 4.3 stated above. \( \square \)
4.6.2 The load balancing process

The process $\Omega_{\text{ALoad}(k)}$ is defined on partially oriented partial pre-allocations $\hat{B}(d, j, n-r, s)$ with $(d, j) \in \hat{L}_{k+1}^r$, as per Definition 4.15. The points in vertex $n+1$ represent all free points in heavy vertices of partially oriented pseudographs with low load-degree sequence $(d, j)$.

Recall Definition 4.1 for the priority function $\text{Pri}_k(d, j)$, and recall that a vertex has highest priority when its priority has the lowest value of all vertices in the partial pre-allocation.

Let

$$L_{l_1,l_2}^d := \{(h_1, \ldots, h_d) : l_1 \leq h_i < l_2, i = 1 \ldots, d\},$$

and $q_{\nu, s, l}$ is given by (4.17).

Ultimately, in Chapters 6 and 7, we shall analyse the load balancing process starting from a $(k+1)$-core, rather than a pseudograph with the distribution of $\mathcal{M}(n, m)$. In combination with the result obtained from the analysis of the $k$-core process, this will lead to our main results on the performance of $\text{Load}(k)$. We use $L_{k+1,2k+1}^d$ as the set of low (less than $2k+1$) degree sequences for $(k+1)$-cores, and we define $\text{ALoad}(k)$ to start with a $(k+1)$-core with low degree sequence in $L_{k+1,2k+1}^d$.

**Algorithm 4.24 ALoad(k)**

Fix $n$ and $m$.

Fix $d^{(0)} \in L_{k+1,2k+1}^r$ with $r < n$ and $2m - \sum_i d_i^{(0)} > (2k+1)(n-r)$.

Let $r(0) = r$ and $s(0) = 2m - \sum_i d_i^{(0)}$.

Let $\hat{B}(0) = \hat{B}(d^{(0)}, 0, n-r(0), s(0))$.

Repeat the following:

Choose vertex $i$ in $\hat{B}(t)$ of highest priority w.r.t. $k$, u.a.r.

Expose a point $e_1$ in $i$.

Expose the mate $e_2$ of $e_1$.

If $e_2$ is in vertex $n+1$, then with probability $q_{n-r(t), s(t), 2k+1}$,
Move $e_2$ and $2k$ unlabelled points from vertex $n+1$ into vertex $r(t)+1$.

$s(t+1) = s(t) - (2k + 1)$.

$r(t+1) = r(t) + 1$.

Make $e_1$ an in-point and $e_2$ an out-point.

$t \leftarrow t + 1$.

Until all vertices 1 to $n$ of $\hat{B}(t)$ have priority greater than $k+1$.

Output $\hat{B}(t)$.

End.

Note that

$$s(t) \geq (2k + 1)(n - r(t)) \tag{4.22}$$

follows from (4.20), so $\textbf{ALoad}(k)$ is well defined. Note also that $r(t)$ increases by at most 1 in each step. If $r(t) = n - 1$ then

$$q_{n-r(t),s(t),2k+1} = \frac{(2k + 1)m_{2k+1,1,s(t),2k+1}}{s(t)} = \begin{cases} 1 & \text{if } s(t) = 2k + 1, \\ 0 & \text{otherwise.} \end{cases}$$

So $r$ will remain at $n - 1$ until $s = 2k + 1$, then the next time when $e_2$ is in $n+1$ both $n - r$ and $s$ will become zero.

Hence

$$r(t) = n \quad \text{if and only if} \quad s(t) = 0. \tag{4.23}$$

Let $\Omega_{\text{ALoad}(k)} := \{\hat{B}(t)\}_{t \geq 0}$. Define $T_s$ to be the time when $\textbf{ALoad}(k)$ ends. Equivalently, $T_s$ is the random time at which $\Omega_{\text{ALoad}(k)}$ becomes constant.

$$T_s := \min\{t : \text{all vertices 1 to } n \text{ of } \hat{B}(t) \text{ have priority greater than } k + 1\}. \tag{4.24}$$

Let $T_f := m$, half the number of points in $\hat{B}(0)$. Two points are processed in each step, so $T_f$ is an upper bound for $T_s$.

The following lemma summarises the correspondence between $\Omega_{\text{ALoad}(k)}$ and $\textbf{Load}(k)$. In particular, we say that $\textbf{ALoad}(k)$ has a successful output if there are no free points
in $\hat{B}(T_s)$. This corresponds to a pseudograph in which all edges are oriented and the maximum in-degree on any vertex is $k$.

Otherwise, if there are free points remaining, this corresponds to a pseudograph which has edges that cannot be oriented without causing the in-degree of a vertex to exceed $k$.

**Lemma 4.25** Assume that $\hat{B}(0)$ is the partial pre-allocation associated with the $(k+1)$-core of $M \in \mathcal{M}(n,m)$. Then $\hat{B}(T_s)$ has the same distribution as the partially oriented partial pre-allocation associated with the output, $M_f$, of $\text{Load}(k)$ applied to $M$. In particular, $M_f$ has $F/2$ undirected edges, where $F$ is a random variable for the number of free points in $\hat{B}(T_s)$.

**Proof.** Let $M_{k+1}$ be the $(k+1)$-core of $M$, and condition on $M_{k+1} \in \mathcal{N}_0 = \mathcal{M}_{2k+1}(d^{(0)}, 0, n-r_0, s_0)$ for some $r_0 < n$, $d^{(0)} \in L_{k+1,2k+1}^r$ and $s_0 = 2m - \sum_i d_i^{(0)} > (2k+1)(n-r_0)$. Let $\hat{B}(0) = \hat{B}(d^{(0)}, 0, n-r_0, s_0)$.

It follows from Corollary 4.5 and Definition 4.9 that, restricting $M_f$ to $\hat{D}_{d,j,\nu,s}$, $M_f$ has the distribution of $\mathcal{M}_{2k+1}(d,j,\nu,s)$. Let $\hat{B}_f$ be the partially oriented partial pre-allocation associated with $M_f$. Then for any $s$, $\nu$ and $(d,j) \in \hat{L}_{2k+1}^{n-\nu}$, conditioning on $M_{k+1} \in \mathcal{N}_0$,

$$
\Pr(\hat{B}_f = \hat{B}(d,j,\nu,s)) = \Pr(M_f \in \mathcal{M}_{2k+1}(d,j,\nu,s)) = \sum_{\mathcal{P}(\mathcal{N}_0, \mathcal{N}_{i+1})} T_i T_{E_0(k)}(\nu, \mathcal{N}_i, \mathcal{N}_{i+1}),
$$

in analogy to (4.1), where $\mathcal{P}(\mathcal{N}_0, \mathcal{N})$ is the set of paths in $G_k$ from $\mathcal{N}_0$ to $\mathcal{N}$. Now, using the same argument as for Lemma 4.23, this time applying Lemma 4.19 and noting that the graph determined by $A\text{Load}(k)$ is isomorphic to $G_k$, we conclude that

$$
\Pr(\hat{B}_f = \hat{B}(d,j,\nu,s)) = \Pr(\hat{B}(T_s) = \hat{B}(d,j,\nu,s)).
$$

This is true conditioning on any appropriate $\mathcal{N}_0$, and since $\text{Load}(k)$ finds the $(k+1)$-core as a first step, the first statement of the Lemma follows. The second statement is clear from the first. \qed
Chapter 5

Analysis of the $k$-core process

The analysis of the $k$-core process, $\Omega_{ACore(k)}$, will reveal the threshold for the existence of a $k$-core in a random pseudograph and also, for any $\rho$, the threshold for a random pseudograph to have a $(k + 1)$-core with density greater than $\rho$.

The same properties are shown to hold for random simple graphs. The main theorem regarding the $k$-core of a random graph is as follows. Recall that the uniform random graph $G(n, m)$ is the uniform probability space of $n$-vertex (simple) graphs with $m$ edges.

Recall that, for $k \geq 0$ integer and $\lambda$ a positive real,

$$f_k(\lambda) := e^{\lambda} - \sum_{i=0}^{k-1} \frac{\lambda^i}{i!} = \sum_{i \geq k} \frac{\lambda^i}{i!}.$$  

Let

$$h_k(\mu) := \frac{\mu}{e^{-\mu}f_{k-1}(\mu)} $$ \tag{5.1}

and

$$c_k := \inf\{h_k(\mu) : \mu > 0\}.$$  

We show in Lemma 5.2 that $c_k$ is a positive real and that for $c > c_k$ the equation $h_k(\mu) = c$ has two positive roots (and just one for $c = c_k$). Define $\mu_{k,c}$ to be the larger one.
Theorem 5.1 Let $c > 0$ and integer $k \geq 3$ be fixed. Suppose that $m = cn/2$ and $G \in \mathcal{G}(n,m)$. For $c < c_k$, $G$ has empty $k$-core a.a.s. For $c > c_k$, the $k$-core of $G$ a.a.s. has $e^{-\mu_{k,c}} f_k(\mu_{k,c}) n(1 + o(1))$ vertices and $\frac{1}{2} \mu_{k,c} e^{-\mu_{k,c}} f_{k-1}(\mu_{k,c}) n(1 + o(1))$ edges.

The proof is given in Section 5.4.

5.1 Variables for the process

Let $M_0 \in \mathcal{M}(n,m)$ be a random pseudograph and let $d^{(0)} \in L^0_k$ be the low degree sequence of $M_0$. Let $s_0 = 2m - \sum_i d_i^{(0)}$ and let $B(0) = B(d^{(0)}, n - r(0), s_0)$. Let $c = 2m/n$.

Let $\Omega_{\text{ACore}(k)} = \{B(t)\}_{t \geq 0}$ be the probability space of sequences of partial pre-allocations determined by $B(0)$ and $\text{ACore}(k)$, and let $T_s$ be the (random) time at which $\text{ACore}(k)$ stops and $B(t)$ becomes constant:

$$T_s := \min\{T \geq 0 : B(t) = B(T) \text{ for all } t \geq T\}.$$

We will analyse $\Omega_{\text{ACore}(k)}$ to deduce information about $B(T_s)$ and hence the $k$-core of $M_0$ in terms of $c$. This, in conjunction with Lemma 3.11, will provide a proof of Theorem 5.1.

Let $L(t)$ be the number of free points in $B(t)$. Then $L(0) = 2m = cn$ and $L(t)$ decreases by two in each step of the process. Hence

$$L(t) = cn - 2t. \quad (5.2)$$

Let

$$T_f = cn/2.$$

This is the total number of edges in $M_0$ and so $T_s \leq T_f$.

For $i = 0, 1, \ldots, k - 1$ let $Y_i(t)$ be random variables for the number of vertices of degree $i$ in $B(t)$. So

$$\mathbb{E}(Y_i(0)) = \sum_j \mathbb{P}(d_j^{(0)} = i) = n \mathbb{P}(D_1 = i),$$
where $\mathbf{D} = (D_1, \ldots, D_n)$ is the degree sequence of $M_0$. By Lemma 3.6,

$$P(D_1 = i) = p_{0,c,i}(1 + o(1)) = \frac{c^i}{e^{ci}} + o(1).$$

So

$$E(Y_i(0)) = n \frac{c^i}{e^{ci!}} + o(n)$$

and by the second moment method, a.a.s.

$$Y_i(0) = \frac{nc^i}{e^{ci!}} + o(n).$$

Hence, a.a.s.

$$s_0 = cn - \sum_{i<k} i(ne^{-c}c^i/i! + o(n)) = cn - cn(1 - e^{-c}f_{k-1}(c)) + o(n) = cne^{-c}f_{k-1}(c) + o(n),$$

and

$$r_0 = \sum_{i<k} (ne^{-c}c^i/i! + o(n)) = n(1 - e^{-c}f_k(c)) + o(n).$$

So to obtain an a.a.s. result it suffices to condition on $M_0$ with $s_0 \sim cne^{-c}f_{k-1}(c)$ and $r_0 \sim n(1 - e^{-c}f_k(c))$.

Let $S(t)$ be a random variable for $s(t)$, so $S(t)$ is the number of points in vertex $n + 1$ of $B(t)$ (those points for which the allocation has not been generated), and let $\nu(t)$ be a random variable for $n - r(t)$, the number of vertices for which the allocation has not been generated. So

$$S(t) = cn - \sum_i Y_i$$

$$\nu(t) = n - \sum_i Y_i(t),$$

85
and
\[ S(0) \sim cn e^{-c} f_{k-1}(c) \quad \text{and} \quad \nu(0) \sim ne^{-c} f_k(c). \quad (5.3) \]

Let \( X_l(t) \) be
\[ X_l(t) = \sum_{i=1}^{k-1} i Y_i(t). \]
Then \( X_l(t) \) is a random variable for the number of points in vertices of degree less than \( k \)
and
\[ L(t) = X_l(t) + S(t), \]
so, using (5.2),
\[ X_l(t) = cn - 2t - S(t) \quad (5.4) \]
and
\[ X_l(0) \sim nc(1 - e^{-c} f_{k-1}(c)). \]
The process continues while \( X_l(t) > 0 \), or equivalently, while
\[ S(t) < cn - 2t. \quad (5.5) \]
Hence
\[ T_s = \min \{ t \geq 0 : S(t) = cn - 2t \}. \]

Define \( \lambda(t) \) to be \( \lambda_{k,S(t)/\nu(t)} \), that is, \( \lambda(t) \) is the positive solution of
\[ \frac{S(t)}{\nu(t)} = \frac{\lambda f_{k-1}(\lambda)}{f_k(\lambda)}. \quad (5.6) \]
It follows from Lemma 3.5 (i), (ii) and (v), that for fixed \( \delta > 0 \), \( \lambda > \delta \) exists uniquely
provided
\[ S(t) - \nu(t)k \to \infty. \quad (5.7) \]
Now,
\[ q_{\nu,S,k} = \frac{k \nu m_{k,\nu,S,k}}{S} \]
\[ \sim \frac{k \nu p_{k,S/\nu,k}}{S} \]
\[
\begin{align*}
\kappa \nu^k \\
\frac{S f_k(\lambda) k!}{\nu^k} \\
= \frac{\nu^k}{S f_k(\lambda)(k-1)!},
\end{align*}
\]
and
\[
\frac{S}{\nu} = \frac{\lambda(f_k(\lambda) + \lambda^{k-1}/(k-1)!)}{f_k(\lambda)}
= \frac{\lambda}{f_k(\lambda)(k-1)!}
\sim \lambda + \frac{q_{\nu,S,k} S}{\nu},
\]
so
\[q_{\nu(t),S(t),k} S(t) \sim S(t) - \lambda(t) \nu(t).\] (5.8)

We shall use the differential equation method to deduce a system of continuous functions that closely approximate the behaviour of \(S(t), \nu(t)\) and \(\lambda(t)\). To apply Theorem 1.2, we first write functions expressing the expected change of each variable, per unit time.

### 5.2 Expected changes of the variables

For each variable, \(X\) we use \(\Delta X(t)\) to denote \(X(t+1) - X(t)\), the change to \(X\) in step \(t\) of the process. If \(t \geq T_s\) then each \(\Delta X(t)\) is zero. Assume that \(t < T_s\) in the following. Conditioning on \(\mathcal{H}_t = (B(0), B(1), \ldots, B(t))\), the history of the process up till time \(t\), for each variable \(X\),
\[
\mathbf{E}(\Delta X(t) \mid \mathcal{H}_t) = \sum_{B'} \mathbf{E}(\Delta X(t) \mid \mathcal{H}_t \land B(t+1) = B') \mathbf{P}(B(t+1) = B' \mid \mathcal{H}_t)
= \sum_{B'} \mathbf{E}(\Delta X(t) \mid B(t) \land B(t+1) = B') \mathbf{P}(B(t+1) = B' \mid B(t))
= \sum_{B'} \mathbf{E}(\Delta X(t) \mid B(t) \land B(t+1) = B') \mathbf{T}_{GD(k)}(B(t), B').
\]

The sum is taken over all possible partial pre-allocations, \(B'\). As argued in the proof of Lemma 4.17, \(\mathbf{T}_{GD(k)}(B(t), B')\) is zero except when \(B'\) is the same as \(B(t)\) less 2 points,
Let $R_1$ be the set of vertices of $B(t)$ with degree greater than zero and less than $k$, and let $r_1$ be the size of $R_1$. As in the proof of Lemma 4.17, for fixed $i_1, i_2 \in R_1$, let $B^{(1)}$ be $B(t)$ less one point in vertex $i_1 \in R_1$ and one point in vertex $i_2 \in R_1$. Let $B^{(2)}$ be $B(t)$ less one point in vertex $i_1 \in R_1$ and one point in vertex $n + 1$. Let $B^{(3)}$ be $B(t)$ less one point in vertex $i_1 \in R_1$, and less $k$ points in vertex $n + 1$, and with vertex $r(t) + 1$ containing $k - 1$ points. Let $d = (d_1, \ldots, d_{r(t)})$ be the low degree sequence of $B(t)$.

Conditional on $t < T_s$, we have the following:

\[
\mathbf{E}(\Delta Y_d(t) \mid B(t + 1) = B^{(i)}) = \begin{cases} 
-\delta_{d,d_{i_1}} + \delta_{d,d_{i_2}} - 1 - \delta_{d,d_{i_2}} - 1 & i = 1 \\
-\delta_{d,d_{i_1}} + \delta_{d,d_{i_1}} - 1 & i = 2 \\
-\delta_{d,d_{i_1}} + \delta_{d,d_{i_1}} + \delta_{d,k-1} & i = 3,
\end{cases}
\]

\[
\mathbf{E}(\Delta X_l(t) \mid B(t + 1) = B^{(i)}) = \begin{cases} 
-2 & i = 1 \\
-1 & i = 2 \\
-2 + k & i = 3,
\end{cases}
\]

\[
\mathbf{E}(\Delta S(t) \mid B(t + 1) = B^{(i)}) = \begin{cases} 
0 & i = 1 \\
-1 & i = 2 \\
-k & i = 3,
\end{cases}
\]

\[
\mathbf{E}(\Delta \nu(t) \mid B(t + 1) = B^{(i)}) = \begin{cases} 
0 & i = 1, 2 \\
-1 & i = 3,
\end{cases}
\]

where $\delta_{i,j}$ is the Kronecker delta. Note that the changes in $X_l$, $S$, $\nu$ and $L$ are independent of the choice of $i_1$ and $i_2$. We will not proceed further with an attempt to analyse $Y_d$, since the desired information about the $k$-core can be obtained without it. Since $X_l(t)$ is directly related to $S(t)$ via (5.4), an explicit analysis of $X_l$ shall also be omitted.

We may assume that $L$ is positive and even and $S - \lambda \nu > 0$. Also $S \leq L$, so $\frac{S - \lambda \nu}{L - 1} \leq 2$, and hence

\[
\frac{S - \lambda \nu}{L - 1} = O(1).
\]
Using (4.18) and (4.19) for $T_{GD(k)}(B(t), B^{(i)})$, and with reference to $t$ omitted for convenience, conditional on (5.5) and (5.7),

$$
E(\Delta S \mid \mathcal{H}_t) = -\sum_{i_1 \in R_1} T_{GD(k)}(B, B^{(2)}) - \sum_{i_1 \in R_1} kT_{GD(k)}(B, B^{(3)})
$$

$$
= \sum_{i_1 \in R_1} \left(-\frac{S(1 - q_{e,S,k})}{r_1(L - 1)} - \frac{ksq_{e,S,k}}{r_1(L - 1)}\right)
$$

$$
= -S - (k - 1)Sq_{e,S,k}/(L - 1),
$$

which, by (5.8),

$$
= -S - (k - 1)(S - \lambda \nu)(1 + o(1))
$$

$$
= \frac{-kS + (k - 1)\lambda \nu}{L - 1} + o(1),
$$

$$
= \frac{-kS + (k - 1)\lambda \nu}{L} + O(1/L),
$$

(5.9)

and

$$
E(\Delta \nu \mid \mathcal{H}_t) = -\sum_{i_1 \in R_1} T_{GD(k)}(B, B^{(3)})
$$

$$
= \sum_{i_1 \in R_1} \frac{-Sq_{e,S,k}}{r_1(L - 1)}
$$

$$
= \frac{-S + \lambda \nu}{L} + O(1/L).
$$

(5.10)

### 5.3 System of equations

We now apply Theorem 1.2, to deduce continuous real functions $z_1(x)$ and $z_2(x)$ that approximate $S(t)/n$ and $\nu(t)/n$. $L(t)/n$ shall be approximated by $c-2x$, according to (5.2).

We include a third function $\mu(x)$ that approximates $\lambda(t)$.

For $\epsilon > 0$, let $D = D(\epsilon) \subseteq \mathbb{R}^4$ be the region

$$
D := \{(x, z_1, z_2) : z_1 < c - 2x, z_1 - kz_2 > \epsilon, z_2 > -\epsilon/2k, x > -\epsilon\}.
$$

(5.11)
The first and second constraints come directly from (5.5) and (5.7). The third combined with the second ensures that $z_1 > \epsilon/2$ and consequently $-\epsilon < x < c/2 - \epsilon/4$ when combined with the first and fourth. This implies $\epsilon/2 < z_1 < c + 2\epsilon$ and also $-\epsilon/2k < z_2 < (c + \epsilon)/k$, hence $D$ is bounded.

Let $T_D = T_D(S, \nu)$ be the stopping time for $D$:

$$T_D := \min\{t : (t/n, S(t)/n, \nu(t)/n) \in D\}.$$

Each variable is bounded by constant times $n$, and it’s clear from the definition of $A\text{Core}(k)$ that for each random variable $X$, $|X(t + 1) - X(t)| \leq k$ for all $t < T_D$. Hence, hypothesis (i) of Theorem 1.2 is satisfied.

In accordance with (5.9) and (5.10), let

$$f_1(x, z_1, z_2) = -\frac{kz_1 + (k - 1)\mu z_2}{c - 2x},$$

$$f_2(x, z_1, z_2) = -\frac{z_1 + \mu z_2}{c - 2x},$$

where $\mu = \lambda_{k,z_1/z_2}$, that is, $\mu$ is determined by

$$\frac{z_1}{z_2} = \frac{\mu f_{k-1}(\mu)}{f_k(\mu)} = \mu + \frac{\mu^k}{f_k(\mu)(k - 1)!}.$$  \hspace{1cm} (5.14)

Then for $t < T_D$, we have $t/n < c/2 - \epsilon/4$, so $L(t) > cn - 2(cn/2 - \epsilon n/4) = \epsilon n/2$ and $O(\frac{1}{n}) = o(1)$ as $n \to \infty$. Hence

$$E(\Delta S(t) \mid \mathcal{H}_t) = f_1(t/n, S(t)/n, \nu(t)/n, L(t)/n) + o(1),$$

and similarly for $\nu(t)$, so hypothesis (ii) of Theorem 1.2 is satisfied.

Finally, since $c - 2x$ is bounded away from zero, $f_1$ and $f_2$ are continuous and satisfy a Lipschitz condition within $D$. Hence, hypothesis (iii) of Theorem 1.2 is satisfied, and we conclude that, a.a.s.

$$S(t) = nz_1(t/n) + o(n) \quad \text{and} \quad \nu(t) = nz_2(t/n) + o(n),$$

uniformly for $0 \leq t \leq \min\{\sigma n, T_D\}$, where the $z_i(x)$ are the solutions to the system of differential equations, $z_i'(x) = f_i(x, z_1, z_2)$, for $i = 1, 2$, with initial conditions $z_1(0) =$
$ce^{-c}f_{k-1}(c)$ and $z_2(0) = e^{-c}f_k(c)$, and $\sigma$ is the supremum of those $x$ for which these solutions remain in $D$. These initial conditions are from (5.3), and note that from the definition of $\mu$, $\mu(0) = c$.

First, we shall solve the system of differential equations and then we shall analyse the point to which the resulting functions for $S$, $\nu$ and $\lambda$ are valid.

A third differential equation, for $\mu$, is found by differentiating both sides of (5.14):

$$\frac{d}{dx} \left( \frac{z_1}{z_2} \right) = \frac{d}{d\mu} \left( \mu + \frac{\mu^k}{f_k(\mu)(k-1)!}\right) \frac{d\mu}{dx}.$$

On the left hand side:

$$\frac{d}{dx} \left( \frac{z_1}{z_2} \right) = \frac{z_1' z_2 - z_1 z_2'}{z_2^2} = \frac{1}{c-2x} \left( \frac{-k z_1}{z_2} + (k-1)\mu + \frac{z_1^2}{z_2^2} - \frac{\mu z_1}{z_2} \right),$$

and, using $d(f_k(\mu))/d\mu = f_k-1(\mu)$, on the right hand side:

$$\frac{d}{d\mu} \left( \mu + \frac{\mu^k}{f_k(\mu)(k-1)!}\right) = 1 + \frac{k \mu^{k-1}}{f_k(\mu)(k-1)!} - \frac{\mu^{k-1} f_k-1(\mu)}{f_k(\mu)^2(k-1)!} = 1 + \frac{k}{\mu} \left( \frac{z_1}{z_2} - \mu \right) - \left( \frac{z_1}{z_2} - \mu \right) \frac{z_1}{z_2} = \frac{-1}{\mu} \left( \frac{-k z_1}{z_2} + (k-1)\mu + \frac{z_1^2}{z_2^2} - \frac{\mu z_1}{z_2} \right) = \frac{-c + 2x}{\mu} \frac{d}{dx} \left( \frac{z_1}{z_2} \right).$$

Hence

$$\frac{d}{dx} \left( \frac{z_1}{z_2} \right) = \frac{-c + 2x}{\mu} \frac{d}{dx} \left( \frac{z_1}{z_2} \right) \frac{d\mu}{dx},$$

and so

$$\mu' = \frac{-\mu}{c-2x}, \quad (5.16)$$

which is easily solved to give

$$\frac{\mu^2}{c-2x} \text{ is constant. } \quad (5.17)$$
We also find that
\[
\frac{d}{d\mu} \left( z_2 e^{\mu} f_k(\mu) \right) = \frac{d z_2}{d\mu} \frac{e^{\mu}}{f_k(\mu)} + \frac{z_2 e^{\mu}}{f_k(\mu)} \frac{z_2 f_k(\mu) - z_2 f_k(\mu)}{f_k(\mu)^2} = 0.
\]

Hence
\[
\frac{z_2}{e^{-\mu} f_k(\mu)} \text{ is constant.} \tag{5.18}
\]

Equations (5.17) and (5.18) are essentially equations (5.7) and (5.8) of [31], although the system of equations there is not exactly the same.

Substituting in the initial conditions to (5.17) and (5.18) gives
\[
c - 2x = \frac{\mu^2}{c},
\]
\[
z_2 = e^{-\mu} f_k(\mu),
\]
and by (5.14),
\[
z_1 = \mu e^{-\mu} f_{k-1}(\mu).
\]

Hence, (5.15) becomes
\[
S(t) = n \mu e^{-\mu} f_{k-1}(\mu) + o(n) \quad \text{and} \quad \nu(t) = n e^{-\mu} f_k(\mu) + o(n), \tag{5.19}
\]
where \( \mu = \mu(t/n) \) is the non-negative root of \( \mu^2 = c(c - 2t/n) \).

To see if (5.19) are valid at time \( T_s \), we need to analyse \( \sigma \), the value of \( x \) for which the solutions first leave \( D \). When \( x = \sigma \) the solution has reached one of the boundaries of \( D \). It cannot be either of the boundaries determined by the third and fourth constraints, since these do not correspond to feasible values of \( t \) or \( \nu \), which are always non-negative.

Hence, (5.15) remain valid until either the first or second constraint of \( D \) is violated.
Let $\sigma_1 := \min\{x \geq 0 : z_1 = c - 2x\}$, and let $\sigma_2 := \min\{x \geq 0 : z_1 - kz_2 = \epsilon\}$.

Note that, in the limit as $x \to c/2$ it’s clear from the solutions that $\mu \to 0$ and $z_1, z_2 \to 0$ also. Hence $\sigma_1 \leq c/2$ and $\sigma_2 < c/2$ exist, and $\sigma = \min\{\sigma_1, \sigma_2\}$.

For non-negative $\mu$, the following are equivalent:
\[
\begin{align*}
    c - 2x &= z_1 \\
    \frac{\mu^2}{c} &= \mu e^{-\mu} f_{k-1}(\mu) \\
    c &= \frac{\mu}{e^{-\mu} f_{k-1}(\mu)} \quad \text{or} \quad \mu = 0.
\end{align*}
\]

Recall (5.1), that
\[
h_k(\mu) := \frac{\mu}{e^{-\mu} f_{k-1}(\mu)}.
\]

Then $\sigma_1 = \min\{c/2, \{x \geq 0 : h_k(\mu) = c\}\}$.

**Lemma 5.2** For positive $\mu$ and $k > 2$, $\lim_{\mu \to 0} h(\mu) = \infty = \lim_{\mu \to \infty} h(\mu)$ and $h(\mu)$ has a unique minimum, which is positive.

**Proof.** Clearly $h_k(\mu)$ is positive for positive $\mu$. As $\mu \to 0$, for $k > 2$,
\[
h_k(\mu) = \sum_{i \geq k-1} \frac{\mu^i}{i!} \to \infty.
\]

On the other hand, $\lim_{\mu \to \infty} e^{-\mu} f_{k-1}(\mu) = 1$ so $\lim_{\mu \to \infty} h(\mu) = \infty$. For $\mu > 0$,
\[
h_k'(\mu) = \frac{e^\mu}{f_{k-1}(\mu)} \left(1 + \mu + \frac{\mu f_{k-2}(\mu)}{f_{k-1}(\mu)}\right).
\]

This is zero if and only if
\[
f_{k-1}(\mu) = \mu^{k-1}/(k-2)!,
\]
which has only one positive solution. \[\square\]

Define
\[
c_k := \inf\{h_k(\mu) : \mu > 0\}.
\]

Then by Lemma 5.2, $c_k$ is a positive real, and for $c > c_k$ the equation $h_k(\mu) = c$ has two positive roots (and just one for $c = c_k$). Define $\mu_{k,c}$ to be the larger one:
\[
\mu_{k,c} := \max\{\mu : h_k(\mu) = c\},
\]

93
when it exists.

First suppose that $c < c_k$. Then $h_k(\mu) = c$ has no solutions in $D$, so $\sigma_1 = c/2 > \sigma_2$ and $\sigma = \sigma_2$. At this point, $z_1 - kz_2 = \epsilon$, which, by (5.14) implies $\psi_k(\mu) - k = O(\epsilon)$. By Lemma 3.5, this implies $\mu = O(\epsilon)$, and so, by (5.19), $\nu = O(\epsilon)$ also.

Hence, by Lemma 4.23, the $k$-core of $M_0$ is a.a.s. of size at most $O(\epsilon)$ vertices.

Now suppose that $c > c_k$, and so $\mu_{k,c}$ exists. Note that, for $\mu > 0$ and $k \geq 2$, $e^\mu > f_{k-1}(\mu)$ and so $h_k(\mu) \geq \mu$. As $\mu(0) = c$ and $\mu'$ is negative, $h_k(\mu(x)) = c$ has a solution for some $0 \leq x < c/2$. Hence, $\sigma_1 < c/2$ exists and $\mu(\sigma_1) = \mu_{k,c}$. This occurs before $\mu$, $z_1$ and $z_2$ approach zero, so for sufficiently small $\epsilon$, $\sigma_1 < \sigma_2$ and $\sigma = \sigma_1$. At this point, by (5.15), $S(t) \sim cn - 2t$, however, with some probability, $T_s > \sigma n$ and the values of the random variables at the end of the process are not within $D$ (note that $T_s = T_D$ at the border determined by the first constraint). To get around this we apply Theorem 1.3, with $\hat{D}$ the same as (5.11), and $D$ replaced by

$$\hat{D} := \{ (x, z_1, z_2) : z_1 - k z_2 > \epsilon, z_2 > -\epsilon/2, -\epsilon < x < c/2 - \epsilon/4, z_1 < c + 2\epsilon \}.$$ 

Note that $\hat{D} \subseteq \tilde{D}$. Let $\tilde{\sigma}$ be the minimum value of $x$ for which the solution leaves $\tilde{D}$. The conclusion of Theorem 1.3 essentially tells us that the solution may be extended beyond the boundary of $\tilde{D}$ into $\hat{D}$, and that it remains valid for $t \leq \min\{\tilde{\sigma} n, T_{\tilde{D}}\}$, provided hypothesis (iii) of Theorem 1.2 holds in the larger domain, $\hat{D}$.

Since $c - 2x$ is bounded away from zero in $\hat{D}$, $f_1$ and $f_2$ are continuous and satisfy a Lipschitz condition within $\hat{D}$. Hence, hypothesis (iii) is satisfied.

For $x \geq \sigma_1$, $h_k(\mu) - c$ begins to go negative, so $c - 2x - z_1$ must do also. By (5.15) it follows that (5.5) must be violated a.a.s. and $t = T_s$ at some $t \sim \sigma n$ a.a.s.

Thus, by Lemma 4.23 the $k$-core of $M_0$ is a.a.s. non-empty and of size and average degree determined by (5.19) at $\mu = \mu_{k,c}$. Let $\rho$ be the density of the $k$-core of $M_0 \in \mathcal{M}(n, cn^2)$. Then a.a.s.

$$\rho = \psi_k(\mu_{k,c})(1 + o(1)). \quad (5.20)$$
5.4 Proof of Theorem 5.1

We have shown that for $c < c_k$, the $k$-core of a multigraph $M \in \mathcal{M}(n, m)$, with $m = cn/2$, is a.a.s. of size at most $O(\epsilon)$ vertices, which by Lemma 3.11, implies that the same can be said of the $k$-core of a random graph $G \in \mathcal{G}(n, m)$. However, Luczak proved in [28] that for every $k \geq 3$, the $k$-core of a random graph $\mathcal{G}(n, m)$ a.a.s. is either empty or contains at least $0.0002n$ vertices. Since $\epsilon$ is arbitrarily small, we conclude that for $c < c_k$, the $k$-core of $\mathcal{G}(n, m)$ is a.a.s. empty.

On the other hand, for $c > c_k$, we have shown that the $k$-core of $M \in \mathcal{M}(n, m)$ is a.a.s. non-empty and of size $e^{-\mu_{k,c}}f_k(\mu_{k,c})n + o(n)$ vertices and $\frac{1}{2}\mu_{k,c}e^{-\mu_{k,c}}f_{k-1}(\mu_{k,c})n + o(n)$ edges. Hence, by Lemma 3.11, the same can be concluded for $G \in \mathcal{G}(n, m)$.
Chapter 6

Analysis of the load balancing process

The aim of the analysis of the load balancing process, $\Omega_{\text{ALoad}(k)}$, is to show that a.a.s $\text{Load}(k)$ will have a successful output on a pseudograph if the average degree, or density, of the $(k+1)$-core of the pseudograph is less than $2^k$. This is done by analysing $\text{ALoad}(k)$ applied to a $(k+1)$-core and showing that a.a.s. the output is successful if the density is initially less than $2^k$. By combining this result with the threshold result for the $(k+1)$-core to have a given density, we deduce the threshold for the success of $\text{Load}(k)$.

Our main theorem and corollary regarding the performance of the load balancing algorithm are as follows.

Let $\lambda_k$ be the solution to

$$\frac{\lambda f_k(\lambda)}{f_{k+1}(\lambda)} = 2^k,$$

and let $\rho_k$ be

$$\rho_k := h_{k+1}(\lambda_k),$$

with $h_l(\lambda)$ defined by (5.1). Then by Theorem 5.1, $\rho_k$ is the threshold density for a random graph to have a $(k + 1)$-core with density $2^k$. 

96
Recall that the uniform random graph $\mathcal{G}(n,m)$ is the uniform probability space of $n$-vertex (simple) graphs with $m$ edges.

**Theorem 6.1** Let $\rho > 0$, $\epsilon > 0$ and integer $k \geq 2$ be fixed. Suppose that $m = \rho n/2$ and let $M$ be a random pseudograph with the distribution of $\mathcal{M}(n,m)$. A.a.s. if $\rho < \rho_k - \epsilon$ then $\text{Load}(k)$ applied to $M$ has a successful output, and a.a.s. if $\rho > \rho_k + \epsilon$, then $\text{Load}(k)$ does not have a successful output. Moreover, the same result holds if $M$ is replaced by $G \in \mathcal{G}(n,m)$.

The proof will follow from Lemma 6.3 and is given in Section 7.3.

It follows from Lemma 1.1 that an optimal edge orienting algorithm, aiming for maximum in-degree no greater than $k$, will not have a successful output on a graph with a $(k+1)$-core (or any subgraph) of density greater than $2k$. Hence, the threshold for an optimal algorithm to have a successful output is no greater than the threshold for the $(k+1)$-core to have density $2k$. But, by Theorem 6.1, the latter is the threshold for the success of $\text{Load}(k)$, so, as a corollary to Lemma 1.1 and Theorem 6.1, we may describe $\text{Load}(k)$ as a.a.s. optimal.

Lemma 1.1 and Theorem 6.1 also imply the following general result on the structure of the random graph.

**Corollary 6.2** Let $G \in \mathcal{G}(n,m)$, and $k > 0$ a fixed integer. The following three properties of $G$ share the same threshold, $\rho_k$:

(i) The $(k+1)$-core of $G$ has average degree less than $2k$.

(ii) $G$ does not contain a subgraph of average degree greater than $2k$.

(iii) there exists an orientation of the edges of $G$ such that the maximum in-degree is no greater than $k$.

In fact, our analysis will imply the slightly stronger result that a.a.s. if the average degree of the $(k+1)$-core of $G$ is less than $2k$ then (ii) and (iii) hold.
Assume that $k$ is fixed in all of the following.

Let $M \in \mathcal{M}(n_0, m_0)$ for some $n_0$ and $m_0$, and let $M_0$ be the $(k+1)$-core of $M$. Then, restricting to $M_0$ having $n \leq n_0$ vertices and $m \leq m_0$ edges, $M_0$ has the distribution of $\mathcal{M}_{k+1}(0, n, m)$.

Let $d^{(0)} \in L^r_{k+1, 2k+1}$ be the low degree (less than $2k+1$) sequence of $M_0$. Let $s(0) = 2m - \sum_i d_i^{(0)}$ and let $\hat{B}(0) = \hat{B}(d^{(0)}, 0, n-r, s(0))$, so that $\hat{B}(0)$ is the partial pre-allocation corresponding to the $(k+1)$-core of $M$. Assume that $r < n$ and $s(0) > (2k+1)(n-r)$.

Let $\rho = 2m/n$, the density of $M_0$.

Let $\Omega_{\text{ALoad}(k)} = \{\hat{B}(t)\}_{t \geq 0}$ be the probability space of sequences of partially oriented partial pre-allocations determined by $\hat{B}(0)$ and $\text{ALoad}(k)$, and let $T_s$ be the (random) time at which $\text{ALoad}(k)$ stops and $\hat{B}(t)$ becomes constant:

\[ T_s := \min\{T \geq 0 : \hat{B}(t) = \hat{B}(T) \text{ for all } t \geq T\}. \]

We will analyse $\Omega_{\text{ALoad}(k)}$ to deduce information about $\hat{B}(T_s)$ and hence the probability that $\text{Load}(k)$ is successful when applied to $M$.

In particular, we shall show the following.

**Lemma 6.3** Fix $\epsilon > 0$ and integer $k \geq 2$. Let $M \in \mathcal{M}(n_0, m_0)$ and let $M_c$ be the $(k+1)$-core of $M$. Let $\rho$ be the density of $M_c$, and let $\hat{B}$ be the partial pre-allocation corresponding to $M_c$. A.a.s. if $\rho < 2k - \epsilon$ then $\text{ALoad}(k)$ applied to $\hat{B}$ will have a successful output.

The proof will not rely specifically on $M_c$ being the $(k+1)$-core, but will rely on the assumption that $M_c$ contains no vertices of degree $k$ or less. We will also use the truncated multinomial distribution of the degrees of vertices in $M_c$ for part of the proof.
6.1 Variables for the process

Let $D$ be the set of possible load-degree, in-degree pairs for vertices in an allocation subjected to $\text{ALoad}(k)$:

$$D := \{(d, j) \in \mathbb{Z}^2 : 0 \leq d \leq 2k + 1, 0 \leq j \leq \lfloor \frac{d}{2} \rfloor \} \setminus \{(2k + 1, 0)\}.$$ 

For $t > 0$, for each $(d, j) \in D$, let $Y_{d,j}(t)$ be a random variable for the number of vertices in $\hat{B}(t)$ with load-degree $d$ and in-degree $j$. Each such vertex contains $d - 2j$ free points. Let $Y(t)$ be a vector of the $Y_\alpha(t)$, for some fixed ordering of the $\alpha$. Let $W(t)$ be a random variable denoting $n - r(t)$, the number of vertices in $\hat{B}(t)$ for which the degree sequence has not been generated. Let $H(t)$ be a random variable for $s(t)$, the number of points in vertex $n + 1$ of $\hat{B}(t)$. Let $L(t)$ be the number of free points in $\hat{B}(t)$. Then

$$L(t) = 2m - 2t = \rho n - 2t,$$  \hspace{1cm} (6.1)

as in (5.2). Note also that

$$H(t) \geq (2k + 1)W(t),$$

and

$$W(t) = 0 \quad \text{if and only if} \quad H(t) = 0,$$

by (4.22) and (4.23).

From the definition of $\text{ALoad}(k)$, For $d = k + 1, \ldots, 2k$, $Y_{d,0}(0)$ is the number of vertices of load-degree $d$ in $\hat{B}(0)$. For all other $\alpha \in D$, $Y_\alpha(0) = 0$. Also,

$$L(0) = 2m,$$

$$W(0) = n - \sum_d Y_{d,0}(0) = n - r,$$

$$H(0) = 2m - \sum_d dY_d = s(0).$$

Let $A$, $B$ and $C$ be the following subsets of $D$. $A$ is the set of load-degree, in-degree pairs for priority 1 vertices, $C$ is for vertices of load-degree $2k + 1$, which will not appear until
the minimum load-degree reaches $2k$, and $B$ is for the rest of the vertices (other than vertex $n+1$).

\[
A := \{ (d, j) \in D : d < 2k \text{ and } d - j \leq k \},
\]
\[
B := \{ (d, j) \in D : k < d < 2k \text{ and } d - j > k \},
\]
\[
C := \{ (d, j) \in D : d = 2k + 1 \}.
\]

Define random variables $X_l(t)$, $X(t)$ and $Z(t)$ as follows:

\[
X_l(t) := \sum_{(d,j) \in A} (d - 2j)Y_{d,j}(t), \quad (6.2)
\]
\[
X(t) := \sum_{(d,j) \in B} (2k - d)Y_{d,j}(t) + X_l(t), \quad (6.3)
\]
\[
Z(t) := H(t) - 2kW(t) + \sum_{(d,j) \in C} Y_{d,j}. \quad (6.4)
\]

$X_l$ counts the number of points in priority 1 vertices, all of which will be processed before $\text{ALoad}(k)$ ends. $X$ counts the maximum number of free points in vertices $1 \ldots r$ (not including any of load-degree greater than $2k$) which could become in-points without the load-degree of any of these vertices exceeding $2k$. $Z$ counts the minimum number of points that need to become out-points so that, after all the points in vertex $n+1$ are allocated to vertices $r+1 \ldots n$, no vertex has load-degree greater than $2k$.

Note that $X_l(0) = 0$ and $Z(0) = H(0) - 2kW(0)$.

**Definition 6.4** The following times $T_h$ and $T_f$ are defined for the process:

\[
T_f := m,
\]
\[
T_h := \min\{ t : X(t) = 0 \}.
\]

$L(T_f) = 0$, so $T_f$ is the time at which all points have been exposed. When $\text{ALoad}(k)$ stops at time $T_s$, there are no vertices of load-degree less than $2k$, so $X(T_s) = 0$ and $T_h \leq T_s \leq T_f$. If $T_h \neq T_s$ then $T_h$ is the first time when a vertex of load-degree $2k$ is
processed. ALoad\((k)\) has a successful output if there are no free points in \(\hat{B}(T_s)\), that is, if (and only if) \(T_s = T_f\).

Note that we may write
\[
X(t) = \sum_{D \setminus C} \min\{2k - d, d - 2j\} Y_{d,j}(t),
\]
and we find the following.

**Lemma 6.5** For any \(0 \leq t_0 \leq T_h\),
\[
T_h \geq t_0 + \frac{1}{2} X(t_0).
\]

**Proof.** Let \(v, 1 \leq v \leq r(t_0)\) be a vertex in \(\hat{B}(t_0)\) and let \(d_0 = d^L(v)\) and \(j_0 = d^-(v)\) in \(\hat{B}(t_0)\). For \(t > t_0\) let \(d_t = d^L(v)\) and \(j_t = d^-(v)\) in \(\hat{B}(t)\). Let \(i_t\) and \(u_t\) be the number of in-points and out-points respectively, appearing in \(v\) between time \(t_0\) and time \(t\). Hence, \(d_t = d_0 + i_t - u_t\) and \(j_t = j_0 + i_t\), so \(d_t - 2j_t = d_0 - 2j_0 - i_t - u_t\). If \(t = T_h\) then either \(d_t = 2k\) or \(d_t - 2j_t = 0\), so
\[
\begin{align*}
    i_t - u_t &= 2k - d_0 \\
    \text{or} \quad i_t + u_t &= d_0 - 2j_0,
\end{align*}
\]
and so
\[
i_t + u_t \geq \min\{2k - d_0, d_0 - 2j_0\}.
\]
Summing over all \(v, 1 \leq v \leq r(t_0)\) gives, on the left hand side, \(2(T_h - t_0)\), since this is the number of points processed from time \(t_0\) to \(T_h\), and on the right hand side, \(X(t_0)\), by 6.5.

Thus at any time, \(X\) is a lower bound for the number of points which will be processed and for twice the number of steps remaining before ALoad\((k)\) ends, so in some sense represents the potential for ALoad\((k)\) to have a successful output. Clearly ALoad\((k)\) will continue as long as \(X\) is positive.
On the other hand, $Z$ represents the potential for $\text{ALoad}(k)$ to not have a successful output. As stated above, at least $Z$ points in heavy vertices and vertex $n + 1$ need to become out-points for there to be no more vertices of load-degree greater than $2k$. If $Z$ is too large, this will not be possible. Note that, for $t < T_h$, there are no vertices of load-degree $2k + 1$, so the last term in (6.4) is zero and $Z(t) = H(t) - 2kW(t)$.

We shall show, in fact, that a.a.s. $\text{ALoad}(k)$ will have a successful output if $X(0)$ is bigger than $Z(0)$.

### 6.1.1 Outline of the proof of Lemma 6.3

In Lemma 6.6, we show that $X(0) - Z(0) = n(2k - \rho)$, so the hypothesis of Lemma 6.3 can be changed from “If $\rho < 2k - \epsilon$” to “If $X(0) > Z(0) + \epsilon n$”. The proof has two main parts. Firstly, in the remainder of Chapter 6 we show that for any constants $\xi$, $\epsilon > 0$, there exist constants $\epsilon_c$, $\omega > 0$ such that

$$\text{a.a.s. if } X(0) > Z(0) + \epsilon n \text{ then } X(t) > \xi Z(t) + \epsilon_c n \text{ at some } t < T_f - \omega n. \quad (6.6)$$

Then, in Chapter 7, we use quite different arguments to show that this conclusion implies that $\text{ALoad}(k)$ has a successful output (Lemma 7.1).

The proof of (6.6) is somewhat involved follows from three lemmas. First of all, we identify a particular event that, if it occurs, indicates a change in the behaviour of the variables describing the process. The time $T_l = T_l(\epsilon_b)$, for some $\epsilon_b > 0$, is defined as the time when this event first occurs (Definition 6.13).

In Lemma 6.8 we analyse how $X(t)$ and $Z(t)$ change in each step of the process, and see that in most instances the change in $X - Z$ is zero. Then, in Section 6.2, we use a branching process argument to show that before time $T_l$ the total change in $X - Z$ is very small. Since both $X$ and $Z$ are decreasing, this implies that the ratio $X/Z$ is, essentially, increasing. Because we are talking about events that occur a.a.s., and events that may not occur at all, the precise statements are a little involved. However, the main conclusion of this section, Lemma 6.15, tells us that, assuming $X(0) > Z(0) + \epsilon n$, a.a.s. either $T_l$ is
reached or $X/Z$ becomes arbitrarily large before the end of the process: For any constants $\xi, \epsilon, \epsilon_b > 0$, there exist constants $\epsilon_c, \omega > 0$ such that

\[
\text{a.a.s. if } X(0) > Z(0) + \epsilon n \text{ then either } T_l(\epsilon_b) < T_f - \omega n \text{ or } X(t) > \xi Z(t) + \epsilon_c n
\]

for some $t < T_f - \omega n$.

Next, in Section 6.3, we use the differential equation method to show that, subject to a particular condition being satisfied at some point early enough in the process, we may conclude from equations describing the behaviour of the variables that the ratio $X/Z$ becomes arbitrarily large before either $X$ or $Z$ gets close to zero. $T_g = T_g(\epsilon_a)$, for some $\epsilon_a > 0$, is defined as the time when the particular condition is first satisfied (Definition 6.16), and Lemma 6.17 states: For any constants $\xi, \omega_a, \epsilon_a > 0$, there exist constants $\epsilon_c, \omega_b > 0$ such that

\[
\text{a.a.s. if } T_g(\epsilon_a) < T_f - \omega_a n \text{ then } X(t) > \xi Z(t) + \epsilon_c n \text{ for some } t < T_f - \omega_b n.
\]

Finally, in Section 6.4 we show Lemma 6.18, which states that: Assuming $X(0) > Z(0) + \epsilon n$ for some $\epsilon > 0$,

\[
\text{a.a.s. if there exist sufficiently small constants } \epsilon_b, \omega > 0 \text{ such that } T_l(\epsilon_b) < T_f - \omega n \text{ then there exists } \epsilon_a > 0 \text{ such that } T_g(\epsilon_a) < T_l(\epsilon_b).
\]

To prove this lemma we analyse some properties of $\Omega_{\text{ALoad}(k)}$ in more detail. We define phases, and show that some of the variables satisfy certain constraints in each phase. With this extra information we deduce functions, $F(k, w, \mu)$ and $D(k, w)$, defined by (6.29) and (6.33), such that if $F(k, w, \mu)$ is greater than $D(k, w)$ for each $k$ and $w$ and at a certain value of $\mu$, then this implies Lemma 6.18. The proof that $F(k, w, \mu) - D(k, w)$ is positive involves finding several continuous lower bounds, each valid in certain parts of the desired range of $k$, $w$ and $\mu$, and showing that these are positive. Most of the proof is relegated to Appendix A.

Hence, Lemmas 6.15, 6.17 and 6.18 together imply (6.6).
In Lemma 6.9 and Corollary 6.10 we show that $A\text{Load}(k)$ will have a successful output if and only if $Z(T_h) = 0$, Hence, we need only analyse $\Omega_{A\text{load}(k)}$ up until time $T_h$ and for the second part of the proof we prove Lemma 7.1, which states: For any $\delta > 0$, there is a constant $\xi = \xi(\delta)$ such that, for any constants $\epsilon_c, \omega > 0$,

a.a.s. if there exists $T < T_f - \omega n$ such that $X(T) > \xi Z(T) + \epsilon_c n$, then

$P(Z(T_h) = 0) > 1 - \delta$.

This is shown in Chapter 7. An outline of the proof is given at the start of that chapter.

6.1.2 Some deterministic features of the algorithm

**Lemma 6.6** Let $B$ be a partial pre-allocation for a $(k+1)$-core with density $\rho$. Let $X$ and $Z$ be defined for $B$. Then

$$\rho = 2k - \frac{1}{n}(X - Z),$$

and

$$\rho < 2k \text{ if and only if } X > Z.$$ 

**Proof.** All vertices have in-degree zero and load-degree greater than $k$, so $X = \sum_{d=k+1}^{2k}(2k-d)Y_{d,0}$. Let $(d_1, d_2, \ldots, d_r)$ be the low degree sequence of $B$. Then

$$X - Z = \sum_{d=k+1}^{2k}(2k-d)Y_{d,0} - (s - 2k(n-r))$$

$$= 2kr - \sum_{i=1}^{r}d_i - s + 2kn - 2kr$$

$$= 2kn - 2m$$

$$= n(2k - \rho),$$

and the result follows. 

For each random variable $Y$, we use $\Delta Y(t)$ denote the change in $Y$ in step $t$ of $A\text{Load}(k)$:

$$\Delta Y(t) := Y(t+1) - Y(t).$$
We define the following events for each step of $\text{ALoad}(k)$.

**Definition 6.7** Suppose that, in step $t$ of $\text{ALoad}(k)$, the edge $f_1f_2$ is exposed and oriented towards $f_1$, with $f_1$ in vertex $i_1$ and $f_2$ in vertex $i_2$. Let $d_1 = d^L(i_1)$, $j_1 = d^-(i_1)$, $d_2 = d^L(i_2)$ and $j_2 = d^-(i_2)$, at time $t$. Define $C_1(t)$, $C_2(t)$ and $C_3(t)$ to be the following events:

\[ C_1(t) = [d_2 \leq 2k \text{ and } d_2 - j_2 > k \text{ at time } t], \]

\[ C_2(t) = [d_2 = 2k + 1 \text{ or } i_2 = n + 1 \text{ at time } t], \]

\[ C_3(t) = [d_2 - j_2 \leq k \text{ at time } t]. \]

Note that exactly one of $C_1(t)$, $C_2(t)$ or $C_3(t)$ must occur in each step of the process. We have the following lemma:

**Lemma 6.8** In the event that $d_1$ is less than $2k$,

- if $C_1(t)$ then $\Delta X(t) = 0$ and $\Delta Z(t) = 0$,
- if $C_2(t)$ then $\Delta X(t) = -1$ and $\Delta Z(t) = -1$,
- if $C_3(t)$ then $\Delta X(t) = -2$ and $\Delta Z(t) = 0$.

In the event that $d_1$ equals $2k$, $C_3(t)$ is not possible since the minimum load-degree is, consequently, $2k$, and

- if $C_1(t)$ then $\Delta X(t) = 1$ and $\Delta Z(t) = 1$,
- if $C_2(t)$ then $\Delta X(t) = 0$ and $\Delta Z(t) = 0$.

**Proof.** Let $\Delta_1 X(t)$ and $\Delta_2 X(t)$ denote the contributions to $\Delta X(t)$ due to $f_1$ and, respectively, $f_2$ being processed, so that $\Delta X(t) = \Delta_1 X(t) + \Delta_2 X(t)$. Similarly define $\Delta_1 Z(t)$ and $\Delta_2 Z(t)$. In step $t$, the load-degree and in-degree of $i_1$ increase by 1 and the load-degree of $i_2$ decreases by 1. Each event and variable in the following is for step $t$, and we drop the argument $t$. There are five cases to show:
If $d_1 < 2k$ then $\Delta_1 X = -1$ and $\Delta_1 Z = 0$.
If $d_1 = 2k$ then $\Delta_1 X = 0$ and $\Delta_1 Z = 1$.
If $C_1$ then $\Delta_2 X = 1$ and $\Delta_2 Z = 0$.
If $C_2$ then $\Delta_2 X = 0$ and $\Delta_2 Z = -1$.
If $C_3$ then $\Delta_2 X = -1$ and $\Delta_2 Z = 0$.

\[
\begin{aligned}
\text{If } d_1 < 2k \text{ then } & \Delta_1 X = -1 \text{ and } \Delta_1 Z = 0. \\
\text{If } d_1 = 2k \text{ then } & \Delta_1 X = 0 \text{ and } \Delta_1 Z = 1. \\
\text{If } C_1 \text{ then } & \Delta_2 X = 1 \text{ and } \Delta_2 Z = 0. \\
\text{If } C_2 \text{ then } & \Delta_2 X = 0 \text{ and } \Delta_2 Z = -1. \\
\text{If } C_3 \text{ then } & \Delta_2 X = -1 \text{ and } \Delta_2 Z = 0.
\end{aligned}
\]

With regard to $Z$, the first, third and fifth statements of $(\ast)$ are true as $f_1$ and $f_2$ contribute nothing to $Z$ either before or after they are processed in these cases. The second statement is true as $i_1$ becomes a vertex of load-degree $2k+1$, which contribute 1 to $Z$. For the fourth case there are three possibilities. If $i_2 \leq n$ and $d_2 = 2k + 1$ then $\Delta_1 Z = \sum C \Delta Y_{d,j} = -1$. If $i_2 = n + 1$ then either $\Delta H = -1$ and $\Delta W = 0$ or $\Delta H = -(2k + 1)$ and $\Delta W = -1$. In both cases $\Delta_1 Z = \Delta H - 2k \Delta W = -1$ as required.

With regard to $X$, if $d_1 < 2k$, by (6.5)

\[
\begin{aligned}
\Delta_1 X &= \min\{2k - (d_1 + 1), d_1 + 1 - 2(j_1 + 1)\} - \min\{2k - d_1, d_1 - 2j_1\} \\
&= \min\{2k - d_1 - 1, d_1 - 2j_1 - 1\} - \min\{2k - d_1, d_1 - 2j_1\} \\
&= -1,
\end{aligned}
\]

so the first statement is true.

The second and fourth statements are true because $f_1$ and $f_2$ contribute nothing to $X$ either before or after they are processed.

For the third and fifth cases, $\Delta_2 X = \min\{2k - (d_2 - 1), d_2 - 1 - 2j_2\} - \min\{2k - d_2, d_2 - 2j_2\}$.

In event $C_1$, $d_2 - j_2 - k \geq 1$ and

\[
\begin{aligned}
\Delta_2 X &= \min\{2k - d_2 + 1, d_2 - 1 - 2j_2\} - (2k - d_2) \\
&= \min\{1, 2(d_2 - j_2 - k) - 1\} \\
&= 1,
\end{aligned}
\]

so the third statement is true.
In event $C_3$, $k - d_2 + j_2 \geq 0$ and

\[
\Delta_2 X = \min\{2k - d_2 + 1, d_2 - 1 - 2j_2\} - (d_2 - 2j_2) = \min\{2(k - d_2 + j_2) + 1, -1\} = -1,
\]

so the fifth statement is true.

Now it’s simply a matter of adding $\Delta_1 X$ to $\Delta_2 X$ and $\Delta_1 Z$ to $\Delta_2 Z$ in each case to get the desired result. □

This leads to the following lemma and corollary.

**Lemma 6.9** For all $t$, $T_h \leq t \leq T_s$, 

\[X(t) \leq 1,\]

and 

\[Z(t) = X(t) + Z(T_h),\]

and $\text{ALoad}(k)$ will have a successful output if and only if $Z(T_h) = 0$.

**Proof.** By Lemma 6.8, $X$ can increase by at most 1 in a single step, and only if $X = 0$. This shows the first part of the lemma, as $X(T_h) = 0$. It follows that at any time $t \geq T_h$ there is at most 1 free point in a priority 1 vertex. If there is one such point, then, because it has, uniquely, the highest priority, in step $t$ it will become an in-point and therefore cannot also become an out-point. Hence event $C_3$ of Lemma 6.8 cannot occur and $\Delta X = \Delta Z$ for all $T_h \leq t \leq T_s$. This implies the second part of the lemma.

At time $T_s$ there are no free points in vertices of load-degree $2k$ or less, and $Z(T_s) = 0$ if and only if there are no free points in vertices of load-degree $2k + 1$ or in vertex $n + 1$. So $\text{ALoad}(k)$ has a successful output if and only if $Z(T_s) = 0$, which by the second part of the lemma, as $X(T_s) = 0$, occurs if and only if $Z(T_h) = 0$. □
Corollary 6.10. \( \text{ALoad}(k) \) will have a successful output if and only if there exists \( T \leq T_h \) such that \( Z(T) = 0 \).

**Proof.** It only needs to be shown that if \( Z(T) = 0 \) for some \( T < T_h \) then \( Z(T_h) = 0 \). The rest is immediate from Lemma 6.9. Assume that such a \( T \) exists. By Lemma 6.8, while \( X > 0, \Delta Z \leq 0 \) and so \( Z(t) = 0 \) for all \( T \leq t \leq T_h \). In particular, \( Z(T_h) = 0 \).

Both \( X \) and \( Z \) are non-increasing if \( X > 0 \) and \( \text{ALoad}(k) \) will have a successful output if \( Z \) goes to zero before \( X \) goes to zero. We shall show that a.a.s. if \( X > Z + \epsilon n \) at \( t = 0 \), for some \( \epsilon > 0 \), then this is the case.

### 6.2 Branching process argument.

In this section we show that, until a certain event occurs, the change in \( X - Z \) is small.

Recall that \( X_l \) counts the number of points in priority 1 vertices and is defined by (6.2). The following function, \( \phi_l(Y(t)) \), is the expected number of free points newly appearing in priority 1 vertices in step \( t \), which we confirm in Lemma 6.12. \( T_l(\epsilon_b) \) is defined as the first time that \( \phi_l(Y(t)) \) becomes larger than \( 1 - \epsilon_b \).

By Lemma 6.8, \( X - Z \) remains constant except when event \( C_3(t) \) occurs. Further, the probability of \( C_3(t) \) depends directly on \( X_l(t) \). We fix \( \epsilon_b > 0 \) and show by a branching process argument that before time \( T_l(\epsilon_b) \), while \( \phi_l(Y(t)) \) is significantly less than 1, \( X_l \) remains small and consequently that the expected number of times \( C_3(t) \) occurs is small.

It follows that the change in \( X - Z \) is also small.

Recall definition (6.1) for \( L(t) \) and the definitions of \( Y_{d,j} \) and \( Y \) from Section 6.1.

**Definition 6.11** Define \( \phi_l(Y) \) to be the following:

\[
\phi_l(Y) := \frac{1}{L} \sum_{d-j=k} (d-2j)(d+1-2j)Y_{d+1,j}.
\]
It’s convenient in the following to make the substitutions \( d = 2k - r \) and \( j = k - r \), and write \( \phi_l(Y) \) as
\[
\phi_l(Y) = \frac{1}{L} \sum_{r=1}^{k} r(r+1)Y_{2k-r+1,k-r}.
\]

**Lemma 6.12** With \( i_2 \) and the event \( C_3(t) \) as in Definition 6.7, let \( D_r(t) \), \( r = 1, \ldots, k \) be the events
\[
D_r(t) := [d^L(i_2) = 2k - r + 1 \land d^L(i_2) - d^r(i_2) = k - r, \text{ at time } t].
\]
We have
\[
\Delta X_l(t) = -I_{X_l(t)>0} - I_{C_3(t)} + \sum_{r=1}^{k} rI_{D_r(t)}, \quad (6.7)
\]
\[
E(\Delta X_l(t)) = -I_{X_l(t)>0} - \frac{X_l(t)}{L(t)} + \phi_l(Y(t)) + O\left(\frac{1}{L}\right). \quad (6.8)
\]

**Proof.** With \( i_1 \) as in Definition 6.7, let \( d_1, d_2, j_1 \) and \( j_2 \) be the load-degrees and in-degrees of \( i_1 \) and \( i_2 \) at time \( t \). Let \( E(t) \) be the event
\[
E(t) := [d_1 - j_1 \leq k, \text{ at time } t].
\]
Event \( E(t) \) is exactly the event that \( i_1 \) is a priority 1 vertex at time \( t \), and the union of the events \( D_r \) is exactly the event that \( i_2 \) becomes a priority 1 vertex upon its load-degree decreasing by one in step \( t \). The changes to \( X_l \) in each of these events are easily seen to be as follows.

Let \( \Delta_1 X_l(t) \) and \( \Delta_2 X_l(t) \) denote respectively, the change in \( X_l \) due to \( i_1 \) being processed and a point in \( i_2 \) being processed in step \( t \), so that \( \Delta_1 X_l(t) + \Delta_2 X_l(t) = \Delta X_l(t) \). We have
\[
\begin{align*}
\text{if } E(t) & \text{ then } \Delta_1 X_l(t) = -1, \\
\text{if } D_r(t) & \text{ then } \Delta_2 X_l(t) = r, \\
\text{if } C_3(t) & \text{ then } \Delta_2 X_l(t) = -1.
\end{align*}
\]
Note that in event \( D_r(t) \), \( r = d_2 - 2j_2 - 1 \), which is the number of free points remaining in \( i_2 \) after a point in it is processed. If \( E(t) \) does not occur then \( \Delta_1 X_l(t) \) is 0. If \( C_3(t) \)
or none of the $D_r(t)$ occur then $\Delta_2 X_l(t)$ is 0. $E(t)$ will occur if there are any priority 1 vertices, that is if $X_l(t) > 0$, so

$$I_{E(t)} = I_{X_l(t) > 0},$$

and (6.7) follows. Note that $P(E(t)) = I_{X_l(t) > 0}$ also.

For each pair $(a, b)$ the probability that $(d_2, j_2) = (a, b)$ is determined by the total number of points in vertices of load-degree $a$ and in-degree $b$ at time $t$. Thus, for each $r = 1, \ldots, k$ the probability that $D_r(t)$ occurs is

$$P(D_r(t)) = P(d_2 = 2k - r + 1 \text{ and } j_2 = k - r) = \frac{1}{L(t)} (r + 1) Y_{2k-r+1,k-r}(t) + O\left(\frac{1}{L}\right).$$

(6.9)

For $C_3(t)$,

$$P(C_3(t)) = \frac{(X_l(t) - 1) I_{X_l(t) > 0}}{L(t) - 1} = \frac{X_l(t)}{L(t)} + O\left(\frac{1}{L}\right).$$

Hence

$$E(\Delta X_l(t)) = -I_{X_l(t) > 0} + \frac{1}{L(t)} \sum_{r=1}^{k} r(r + 1) Y_{2k-r+1,k-r}(t) - \frac{X_l(t)}{L(t)} + O\left(\frac{1}{L}\right)$$

$$= -I_{X_l(t) > 0} + \phi_l(Y(t)) - \frac{X_l(t)}{L(t)} + O\left(\frac{1}{L}\right).$$

Now we define the stopping time, $T_l(\epsilon_b)$, for the event that $\phi_l(Y)$ gets to within $\epsilon_b$ of 1. By (6.8), $\phi_l(Y(t))$ is the expected number of points newly appearing in priority 1 vertices in step $t$. Roughly speaking, if $\phi_l(Y)$ becomes larger than 1, points in priority 1 vertices will start to accumulate and $X_l$ will start to grow. We call this an explosion of priority 1 vertices. For small $\epsilon_b$, the time $T_l(\epsilon_b)$, if reached, indicates that an explosion is (possibly) about to occur.

**Definition 6.13** For $\epsilon_b > 0$, let

$$T_l(\epsilon_b) := \min\{t : \phi_l(Y(t)) > 1 - \epsilon_b\}.$$
Fix constant $\epsilon_b > 0$ for use in the following two lemmas. We write $T_l$ for $T_l(\epsilon_b)$. We now show that before time $T_l$, $X_l$ remains small.

Let $\#C_3(T)$ denote the number of times $t$, $0 \leq t < T$, for which $C_3(t)$ of Definition 6.7 holds.

**Lemma 6.14** Fix $\omega > 0$ and let $T'_l := \min\{T_l, T_f - \omega n\}$. Then

$$A.a.s. \quad \#C_3(T'_l) \leq \log^3 n.$$  

**Proof.** Recall the definitions of a $K$, $c$ tail, an exponential tail and stochastic domination from Section 1.4.5. Define $W(t)$ to be

$$W(t) = \sum_{r=1}^{k} r I_{D_r(t)}.$$  

If $t < T'_l$, $W(t)$ is the number of points in priority 1 vertices newly appearing in step $t$, that is, $W(t)$ is the positive part of (6.7). Otherwise $W(t)$ is zero.

Define $Z_s$ to be, if $X_l(s) = 0$, the size of the branching process (of points in priority 1 vertices) starting at time $s$ and ending at time $\bar{s} > s$ if $X_l(\bar{s}) I_{\bar{s} < T'_l} = 0$, and with $W(t)$ children at each step $s \leq t < \bar{s}$. $Z_s$ is defined to be zero if $X_l(s) > 0$. Fix $s < T'_l$ and condition on $X_l(s) = 0$ for the following. Using (6.9),

$$P(W(t) = r) = I_{t < T'_l} P(D_r(t)) = I_{t < T'_l} \frac{(r + 1) Y_{2k-r+1,k-r}(t) + O(1)}{L(t)}.$$  

For the first $\lfloor \log^2 n \rfloor$ steps of the branching process, or all of them if $\bar{s} < \log^2 n$, the values of $Y_{2k-r+1,k-r}(t)$, for $r = 1, \ldots, k$, change by at most $\lfloor \log^2 n \rfloor$: While $X_l > 0$, the new in-point in each step is in a priority 1 vertex, so $Y_{2k-r+1,k-r}$ can only change if the new out-point is in a vertex counted by $Y_{2k-r+1,k-r}$ or $Y_{2k-r+2,k-r}$, in which case it will change by $\pm 1$. Also, $L$ decreases by 2 in each step and $L(t) > \omega n$ for $t < T_f - \omega n$. So, for
\[ s \leq t < \min\{\bar{s}, \lfloor \log^2 n \rfloor\}, \]

\[ P(D_r(t)) \leq \frac{(r + 1)(Y_{2k-r+1-k-r}(s) + \log^2 n)}{L(s)} \left(\frac{L(s)}{L(t)}\right) \]

\[ < P(D_r(s)) \left(\frac{\omega n + \log^2 n}{\omega n}\right) + \frac{(r + 1) \log^2 n}{\omega n} \]

\[ = P(D_r(s)) + O\left(\frac{\log^2 n}{\omega n}\right) \]

\[ = P(D_r(s)) + \omega n + \log_2 n \omega n \]

\[ = P(D_r(s)) + \omega n + \log_2 n \omega n \]

\[ \leq P(D_r(s)) + O(1). \]

For \( r = 1, \ldots, k \), and some \( \delta < \epsilon_b \), let

\[ P_r := \frac{(r + 1)Y_{2k-r+1-k-r}(s)}{L(s)(1 - \delta)}, \]

and let \( \tilde{W} \) be the random variable with distribution

\[ P(\tilde{W} = r) = \begin{cases} 
P_r & r = 1, \ldots, k \\
1 - \sum_{i=1}^{k} P_i & r = 0 \\
0 & \text{otherwise.} 
\end{cases} \]

Then for \( n \) sufficiently large, and \( s \leq t < \lfloor \log^2 n \rfloor \) (for \( t \geq \bar{s} \), \( W(t) = 0 \) by definition), it’s clear that \( W(t) \) is stochastically dominated by \( \tilde{W} \):

\[ P(W(t) \geq r) \leq P(\tilde{W} \geq r) \quad \text{for } r \geq 0. \]

\( \tilde{W} \) is bounded above by a constant, so has an exponentially small tail, and

\[ E(\tilde{W}) = \sum_{r=1}^{k} rP_r \]

\[ = \sum_{r=1}^{k} \frac{r(r + 1)Y_{2k-r+1-k-r}(s)}{L(s)(1 - \delta)} \]

\[ = \phi(Y(s))/(1 - \delta) \]

\[ < \frac{1 - \epsilon_b}{1 - \delta} < 1. \]

Define \( \tilde{Z} \) to be the size of the branching process with \( \tilde{W} \) children at each step. Applying Theorem 1.4, since \( E(\tilde{W}) < 1 \) by the choice of \( \delta \) and \( \tilde{W} \) has an exponentially small tail, \( \tilde{Z} \) has also, and there exist constants \( K, c > 0 \) such that

\[ P(\tilde{Z} \geq w) \leq Ke^{-cw}. \]
Now, $\tilde{Z}$ stochastically dominates $Z_s$ for $t < \lfloor \log^2 n \rfloor$, so for $C$ and $n$ sufficiently large,

$$P(Z_s \geq C \log n) \leq P(\tilde{Z} \geq C \log n) = O\left(\frac{1}{n}\right),$$

and it follows from this that

$$E(Z_s) = O(\log n).$$

This is uniform for each $s < T'_l$, by fixing $\delta$.

For $s \leq t < \bar{s}$, $X_l(t)$ is the number of free points in the branching process at time $t$ and so is bounded by the total number of points in the branching process, which is $Z_s$: $X_l(t)I_{s \leq t < \bar{s}} \leq Z_s$.

Define $\#C_s$ to be the number of times $C_3(t)$ occurs during the branching process starting at time $s$, assuming $X_l(s) = 0$. $\#C_s$ is defined to be zero if $X_l(s) > 0$. Then, conditioning on $X_l(s) = 0$,

$$E(\#C_s) = \sum_s P(C_3(t) \wedge s \leq t < \bar{s})$$

$$= \sum_t E(X_l(t))I_{s \leq t < \bar{s}}$$

$$\leq \sum_t E(Z_s)I_{s \leq t < \bar{s}}$$

$$= O\left(\frac{\log^2 n}{n}\right).$$

So, without the conditioning

$$E(\#C_3(T'_l)) = \sum_s E(\#C_s I_{t < T'_l})$$

$$\leq \sum_s E(\#C_s | X_l(s) = 0) I_{t < T'_l}$$

$$= O\left(n \frac{\log^2 n}{n}\right)$$

$$= O(\log^2 n).$$

Applying Markov’s inequality,

$$P(\#C_3(T'_l) > \log^2 n) = O\left(\frac{1}{\log^2 n}\right).$$

In the next lemma, we show that, assuming $X(0) > Z(0) + \epsilon n$, a.a.s. either $X/Z$ becomes arbitrarily large, or the stopping time $T_l$ is reached significantly before the end of the
process. This means, essentially, that if there is no explosion of priority 1 vertices, $X/Z$ becomes arbitrarily large.

**Lemma 6.15** For any constants $\xi > 0$ and $\epsilon > 0$, there exists $\epsilon_c$, $\omega > 0$, such that for sufficiently large $n$, if $X(0) > Z(0) + \epsilon n$ then a.a.s. either $T_l < T_f - \omega n$ or $X > \xi Z + \epsilon_c n$ at some time before $T_f - \omega n$.

**Proof.** Let $\omega = \frac{\epsilon}{8\xi}$ and $\epsilon_c = \frac{\epsilon}{4}$. We will argue by contradiction. Firstly, assume that $T_l \geq T_f - \omega n$. By Lemma 6.14 a.a.s.

$$\#C_3(T_f - \omega n) \leq \log^3 n.$$  

Assume that this is true. Let $R = X - Z$. Then $\Delta R(t) = \Delta X(t) - \Delta Z(t)$ at each time $t$. So, by Lemma 6.8,

- if $C_1(t)$ then $\Delta R(t) = 0$,
- if $C_2(t)$ then $\Delta R(t) = 0$,
- if $C_3(t)$ then $\Delta R(t) = -2$.

Hence, at any time $t < T_f - \omega n$,

$$R(t) > R(0) - 2\log^3 n > \epsilon n - 2\log^3 n.$$  

So, assuming $n$ is large enough that $2\log^3 n < \frac{1}{2}\epsilon n$,

$$X(t) - Z(t) > \frac{1}{2}\epsilon n,$$

and in particular,

$$X(t) > \frac{1}{2}\epsilon n \quad (6.10)$$

for all $t < T_f - \omega n$. $X$ is positive, so $T_s \geq T_f - \omega n$ and the process does not become constant before time $T_f - \omega n$. It follows that for $t > T_f - 2\omega n$,

$$2\omega n > L(t) \geq X(t) + Z(t),$$

so

$$Z(t) < 2\omega n.$$
Secondly, assume that $X(t) \leq \xi Z(t) + \epsilon_c n$ for all $t < T_f - \omega n$. In the range $T_f - 2\omega n < t < T_f - \omega n$, we have

$$X(t) \leq \xi Z(t) + \epsilon_c n < 2\xi \omega n + \frac{\epsilon n}{4} = \frac{\epsilon n}{2},$$

which contradicts (6.10).

Thus, a.a.s. if $T_l \geq T_f - \omega n$ then $X(t) > \xi Z(t) + \epsilon_c n$ for some $t < T_f - \omega n$. This implies the lemma.  

### 6.3 Differential equation argument

In the previous section we showed, essentially, that assuming $X(0) > Z(0) + \epsilon n$, a.a.s. if the event characterised by $T_l$ does not occur then there is no explosion of priority 1 vertices, and $X/Z$ becomes arbitrarily large before the end of the process.

Now, in Lemma 6.17, we show that a.a.s. if certain other conditions, characterised by a different stopping time $T_g$ become true, then $X/Z$ will grow arbitrarily large before time $T_s$.

Next, in Section 6.4, we shall show that a.a.s. if $T_l \leq T_s$, then the conditions characterised by $T_g$ are true at time $T_l$. From these three arguments we will be able to conclude that a.a.s., whether there is an explosion of priority 1 vertices or not, $X/Z$ will grow arbitrarily large before the end of the process.

Along similar lines to the proof of Theorem 5.1 in Chapter 5 we will derive equations for the expected changes in each random variable. Then we will deduce a system of differential equations in continuous variables that, on applying Theorem 1.2, we claim approximates the behaviour of our random variables.

Recall the definitions of $H$ and $W$ from the start of Section 6.1. $H(t)$ is a random variable for the number of points, at time $t$, in vertex $n + 1$ of $\hat{B}(t)$, the partial pre-allocation
subjected to $\text{ALoad}(k)$. $W(t)$ is a random variable for the number of vertices of $\hat{B}(t)$ for which the allocation has not yet been generated.

In terms of a pseudograph corresponding to $\hat{B}(t)$, $W$ and $H$ are the numbers of heavy vertices (with in-degree zero) and points in these heavy vertices, respectively.

**Definition 6.16** For $\epsilon_a > 0$, let

$$g(x, z, h) := \frac{x}{z} \left(1 - \frac{2z}{h}\right) - 1$$

and

$$T_g(\epsilon_a) := \min\{t : g(X, Z, H) > \epsilon_a\}.$$  

**Lemma 6.17** For any constants $\xi > 0$, and $\omega_a, \epsilon_a > 0$, there exists $\epsilon_c, \omega_b > 0$ such that a.a.s. if $T_g(\epsilon_a) < T_f - \omega_a n$, then $X > \xi Z + \epsilon_c n$ at some time before $T_f - \omega_b n$.

**Proof.** Fix $\epsilon_a > 0$. In the following we write $T_g$ for $T_g(\epsilon_a)$.

Assume that $T_g < T_f - \omega_a n$. Then there exists $\bar{\omega}_a > \omega_a$ and $\bar{\epsilon}_a > \epsilon_a$ such that $g(X, Z, H) = \bar{\epsilon}_a$ at time $t_0 := T_f - \bar{\omega}_a n$. This implies that $H > 2Z > 0$ and $X > (1 + \epsilon_a)Z$, at this time, so assume that $X(t_0) > 2\epsilon_c n$ for some $\epsilon_c > 0$.

If $Z < (X - \epsilon_c n)/\xi$ at time $t_0$ we are done. So assume that $Z(t_0)$ is at least constant times $n$. Assume also that $(2k + 1)Z - H$ is at least constant times $n$ at time $t_0$. Note that by (4.22), $(2k + 1)Z - H \geq 0$ always. A similar analysis to the following will give the same result in the degenerate case where $(2k + 1)Z - H = 0$ (it can also be shown that the degenerate case a.a.s. does not occur).

When an edge $f_1 f_2$ is processed so that $f_1$ becomes an in-point, the point $f_2$ is uniformly distributed on all $L$ free points remaining.

Hence, provided

$$X(t) > 0,$$  \hspace{1cm} (6.11)

116
in the notation of Lemma 6.8

\[
P(C_1) = \frac{L - H - X_l - I_{X_l=0}}{L - 1} = \frac{L - H - X_l}{L} + O\left(\frac{1}{L}\right),
\]

\[
P(C_2) = \frac{H}{L - 1} = \frac{H}{L} + O\left(\frac{1}{L}\right),
\]

\[
P(C_3) = \frac{(X_l - 1)I_{X_l>0}}{L - 1} = \frac{X_l}{L} + O\left(\frac{1}{L}\right),
\]

with asymptotics as \(n \to \infty\).

If \(t < T_f - \omega_b n\), for fixed \(\omega_b > 0\), then

\[
L(t) = 2m - 2t > 2m - 2(T_f - \omega_b n) = 2\omega_b n, \tag{6.12}
\]

so \(O\left(\frac{1}{L}\right) = o(1)\) as \(n \to \infty\) in this range of \(t\).

Thus, for \(t < T_h\), in step \(t\) of the process, conditioning on the history \(\mathcal{H}_t\),

\[
E(\Delta X \mid \mathcal{H}_t) = E(\Delta X \mid C_1)P(C_1) + E(\Delta X \mid C_2)P(C_2) + E(\Delta X \mid C_3)P(C_3)
\]

\[
= 0 - \frac{H}{L} - 2\frac{X_l}{L} + o(1),
\]

\[
E(\Delta Z \mid \mathcal{H}_t) = E(\Delta Z \mid C_1)P(C_1) + E(\Delta Z \mid C_2)P(C_2) + E(\Delta Z \mid C_3)P(C_3)
\]

\[
= 0 - \frac{H}{L} + 0 + o(1),
\]

using Lemma 6.8. So, including (6.8),

\[
E(\Delta X(t) \mid \mathcal{H}_t) = \frac{-H(t) - 2X_l(t)}{L(t)} + o(1),
\]

\[
E(\Delta X_l(t) \mid \mathcal{H}_t) = -I_{X_l(t)>0} \frac{X_l(t)}{L(t)} + \phi_l(Y(t)) + o(1),
\]

\[
E(\Delta Z(t) \mid \mathcal{H}_t) = \frac{-H(t)}{L(t)} + o(1). \tag{6.13}
\]
The expected change for $H$ is the same as for $S$ in $\Omega_{\text{ACore}(k)}$, given by (5.9), but with $2k + 1$ in place of $k$. Also, we are using $Z = H - 2kW$, rather than $W$, which corresponds to $\nu$ in the analysis of $\Omega_{\text{ACore}(k)}$. So we obtain

$$E(\Delta H(t) \mid H_t) = -\frac{(2k + 1)H(t) + \lambda(t)(H(t) - Z(t))}{L(t)} + o(1), \quad (6.14)$$

where $\lambda$ is the solution of

$$\psi_{2k+1}(\lambda) = \frac{2kH}{H - Z}. \quad (6.16)$$

(Recall that in the analysis of $\Omega_{\text{ACore}(k)}$, $\lambda$ was the solution to the corresponding equation $\psi_k(\lambda) = S/\nu$). For $\alpha > 0$, $\lambda > \alpha$ exists uniquely provided

$$H > Z \quad \text{and} \quad (2k + 1)Z - H \to \infty, \quad (6.15)$$

by the same reasoning as for (5.7).

The equation for $X_l$ relies on some of the $Y_{d,j}$, some of which will be very small or zero at times in the process. $\Omega_{\text{ALoad}(k)}$ apparently goes through a number of phase changes, with regard to the $Y_{d,j}$, with the time at which each phase change occurs being a random variable. This makes it difficult to justify continuous functions that approximate the $Y_{d,j}$ and $X_l$, and consequently $X$, during the process.

Instead, we take advantage of the fact that $X_l \leq X$, and that none of the expected changes for the other variables are dependent on $X$ or $X_l$.

Let $\tilde{X}$ be an artificial random variable modelled on $X$ from time $t_0$, with $\tilde{X} = X$ at time $t_0$ and

$$\Delta \tilde{X} = \frac{-H - 2\tilde{X}}{L} = E(\Delta X) - \frac{2(\tilde{X} - X_l)}{L}. \quad (6.16)$$

If at some time $t$, $\tilde{X}(t) \geq X_l(t)$ then clearly, $E(\Delta \tilde{X}(t)) \leq E(\Delta X(t))$, and so $E(\tilde{X}(t + 1) \mid \tilde{X}(t)) \leq E(X(t + 1) \mid X(t))$.

Conversely, suppose $\tilde{X}(t) = X_l(t) - \eta$ for some $\eta > 0$. Then $E(\Delta \tilde{X}(t)) = E(\Delta X(t)) + \frac{2\eta}{L}$.
and, assuming $L > 2$,

$$
E(\tilde{X}(t+1) \mid \tilde{X}(t)) = \tilde{X}(t) + E(\Delta \tilde{X}(t)) \\
= X(t) - \eta + E(\Delta X(t)) + \frac{2\eta}{L} \\
\leq X(t) + E(\Delta X(t)) - \eta(1 - \frac{2}{L}) \\
< E(X(t+1) \mid X(t)).
$$

Hence, conditioning on the history $\mathcal{H}_t$ of the process, for $t > t_0$,

$$
E(\tilde{X}(t+1) \mid \mathcal{H}_t) \leq E(X(t+1) \mid \mathcal{H}_t).
$$

In accordance with (6.16), (6.13) and (6.14), let

$$
f_1(s, \tilde{x}, z, h, l) = -\frac{h - 2\tilde{x}}{l},
$$

$$
f_2(s, \tilde{x}, z, h, l) = -\frac{h}{l},
$$

$$
f_3(s, \tilde{x}, z, h, l) = -\frac{(2k+1)h + \mu(h - z)}{l},
$$

with

$$
l(s) = l(0) - 2s,
$$

in accordance with (6.1), and $\mu$ the solution to

$$
\psi_{2k+1}(\mu) = \frac{2kh}{h - z}.
$$

The continuous variables $s$, $\tilde{x}$, $z$, $h$ and $l$ are to model $t/n$, $\tilde{X}/n$, $H/n$ and $L/n$ respectively and $\mu$ models $\lambda$.

To apply Theorem 1.2, we first define a domain $D = D(\epsilon)$, $0 < \epsilon < \bar{\omega}_a$, to be

$$
D := \{(s, \tilde{x}, z, h, l) : \tilde{x} > 0, h > z, (2k + 1)z - h > \epsilon, 2\epsilon < l < 2\bar{\omega}_a + 2\epsilon, \\
l > \tilde{x} + h, -\epsilon < s < \bar{\omega}_a\}.
$$

The first four constraints are directly from (6.11), (6.15) and (6.12) (with the inclusion of $2\bar{\omega}_a + 2\epsilon > l$) and ensure that $\tilde{x}$, $z$, $h$ and $l$ are positive in $D$. In particular, the second
and third constraints together imply that \( z, h \) and \( \mu \) are positive and bounded away from zero. The fifth and sixth constraints and \( 2\omega_a + 2\epsilon > l \) ensure that \( D \) is bounded.

These constraints ensure that with respect to \( \tilde{X}, Z, H, L \) and \( f_1, f_2 \) and \( f_3 \), conditions (i), (ii) and (iii) of Theorem 1.2 are satisfied for all \( t < T_D \), the stopping time determined by \( D \).

Consider the system of differential equations, \( S \), determined by

\[
\frac{d\tilde{x}}{ds} = f_1(s, \tilde{x}, z, h), \\
\frac{dz}{ds} = f_2(s, \tilde{x}, z, h), \\
\frac{dh}{ds} = f_3(s, \tilde{x}, z, h).
\]

A fourth equation for \( \mu \) is obtained analogously to (5.16):

\[
\frac{d\mu}{ds} = -\frac{\mu}{l}. \tag{6.22}
\]

We are interested in the solution to the system \( S \) with all variables positive and initial conditions satisfying

\[
l(0) = 2\omega_a > h(0) + \tilde{x}(0),
\]

\[
(2k + 1)z(0) > h(0),
\]

and

\[
g(\tilde{x}(0), z(0), h(0)) = \bar{\epsilon}_a.
\]

It’s clear from our assumptions that setting

\[
\tilde{x}(0) = \frac{X(t_0)}{n}, \quad z(0) = \frac{Z(t_0)}{n}, \quad h(0) = \frac{H(t_0)}{n} \quad \text{and} \quad l(0) = \frac{L(t_0)}{n},
\]

the three conditions above are satisfied and \((0, \tilde{x}(0), z(0), h(0), l(0))\) is in \( D \).

Then applying Theorem 1.2 with these initial conditions, we have a.a.s.

\[
\tilde{X}(t) = n\tilde{x}(t-t_0)/n + o(n), \tag{6.23}
\]

\[
Z(t) = nz(t-t_0)/n + o(n), \tag{6.24}
\]

120
for $0 \leq t \leq \min\{T_D, \sigma_n + t_0\}$, where $\tilde{x}$ and $z$ are solutions of $S$ and $\sigma$ is the supremum of those $s$ to which the solution can be extended in $D$.

We shall show that for $\epsilon$ sufficiently small, $\tilde{x}/z$ becomes arbitrarily large in $D(\epsilon)$, from which we will be able to conclude that a.a.s. $\tilde{X}/Z$ becomes arbitrarily large before the stopping time $T_D$ is reached.

To do this we analyse how the solution to $S$ with given initial conditions approaches the boundary of $D$. We shall show that the solution will remain in $D$ as $s$ increases until either the first, third or fourth constraint in the definition of (6.21) is violated. In other words, the solution does not reach a boundary of $D$ determined by the second or fifth constraints (the sixth trivially won’t be reached). It’s clear that $\tilde{x}, z, l$ and $\mu$ are all decreasing with $s$. Note that $l(0) = 2\bar{\omega}_a$ so $l = 2\bar{\omega}_a - 2s$ and hence the constraint $2\bar{\omega}_a + 2\epsilon$ cannot be violated at the boundary of $D$.

To see that $h$ is decreasing with $s$, note that $\psi_{2k+1}(\mu) > \mu$ by Lemma 3.5 part (iii), so

$$\frac{2kh}{h - z} > \mu,$$

and

$$-(2k + 1)h + \mu(h - z) < 0,$$

and so $f_3$ is negative in $D$. Note also that $(2k + 1)z - h$ is decreasing with $s$:

$$\frac{d}{ds}((2k + 1)z - h) = (2k + 1)f_2 - f_3 = \frac{-\mu(h - z)}{l} < 0.$$

Next, we show that $l > h + \tilde{x}$, so the fifth constraint of (6.21) cannot be violated. This is true at $s = 0$. Suppose that for some some $s$, $l - (\tilde{x} + h) = \eta$ with $0 < \eta < \mu z/2$. Then at this point

$$\frac{d}{ds}(l - (\tilde{x} + h)) = -2 + \frac{h - 2\tilde{x}}{\tilde{x} + h + \eta} + \frac{(2k + 1)h - \mu(h - z)}{\tilde{x} + h + \eta}$$

$$= \frac{-2\eta + (2k - \mu)h + \mu z}{\tilde{x} + h + \eta} > \frac{-2\eta + \mu z}{\tilde{x} + h + \eta} > 0,$$
so \( h + \tilde{x} \) is diverging from \( l \) and must remain less than \( l \).

Lastly, we show that \( h > z \), so the second constraint of (6.21) cannot be violated. At \( s = 0 \), \( z/h < 1/2 \) and we shall show that \( z/h \) is decreasing with \( s \). Now, \( z/h = 1 - 2k/\psi_{2k+1}(\mu) \), and \( \psi_{2k+1}(\mu) \) increases with \( \mu \) by Lemma 3.5 part (v), for all \( \mu > 0 \). Hence

\[
\frac{d}{ds} \left( \frac{z}{h} \right) = \frac{d}{ds} \left( \frac{-2k}{\psi} \right) = \frac{2k}{\psi^2} \frac{d\psi}{d\mu} \frac{d\mu}{ds} = -\frac{2k\mu}{l\psi^2} \frac{d\psi}{d\mu} < 0.
\]

This also implies that

\[
\frac{d}{ds} \left( 1 - \frac{2z}{h} \right) > 0.
\]

Thus, in \( D \), \( (1 - 2z/h) \) is positive and increasing with \( s \), and

\[
\frac{d}{ds} \left( g(\tilde{x}, z, h) \right) = \frac{d}{ds} \left( \frac{\tilde{x}}{z} \right) \left( 1 - \frac{2z}{h} \right) + \frac{d}{ds} \left( 1 - \frac{2z}{h} \right) \frac{\tilde{x}}{z} > 0 \quad \text{if} \quad \frac{d}{ds} \left( \frac{\tilde{x}}{z} \right) > 0.
\]

Combining (6.18) and (6.19),

\[
\frac{d}{ds} \left( \frac{\tilde{x}}{z} \right) = \frac{1}{\tilde{x}} \frac{d\tilde{x}}{ds} \frac{z}{z} - \frac{\tilde{x}}{z} \frac{dz}{ds} = \frac{h}{z^2} \left( -1 - 2\frac{\tilde{x}}{h} + \frac{\tilde{x}}{z} \right) = \frac{h}{z^2} g(\tilde{x}, z, h).
\]

At \( s = 0 \), \( g(\tilde{x}, z, h) > 0 \) so \( \tilde{x}/z \) is increasing. This in turn means that \( g(\tilde{x}, z, h) \) is increasing, so \( \frac{d}{ds} \left( \frac{\tilde{x}}{z} \right) \) is positive always. To see that \( \tilde{x}/z \) is unbounded, consider the derivative of \( \tilde{x}/z \) with respect to \( z \):

\[
\frac{d}{dz} \left( \frac{\tilde{x}}{z} \right) = \frac{d}{ds} \left( \frac{\tilde{x}}{z} \right) \frac{ds}{dz} = -\frac{1}{z} g(\tilde{x}, z, h),
\]

122
which implies $\tilde{x}/z$ goes to infinity as $s$ increases and $z$ decreases to zero. Thus, for any $\alpha > 0$ there is a time $s_\alpha$ such that the solution to $\mathcal{S}$ satisfies $\tilde{x}(s_\alpha) > \alpha z(s_\alpha)$ with all variables positive at $s = s_\alpha$. In particular, $l/2 = \bar{\omega}_a - s_\alpha$ is positive, so $s_\alpha < \bar{\omega}_a$.

By (6.23) and (6.24), a.a.s. the stochastic variables $\tilde{X}(t)/n$ and $Z(t)/n$ closely follow the solutions $\tilde{x}((t-t_0)/n)$ and $z((t-t_0)/n)$ for $t \leq \min\{T_D, \sigma n + t_0\}$.

Let $t_\alpha = ns_\alpha + t_0$. To ensure that $s_\alpha < \sigma$ and the solution has not left $D(\epsilon)$ at this point, we choose an $\epsilon$ satisfying

$$\epsilon < \inf_{0 \leq s \leq s_\alpha} \left\{ \tilde{x}(s), l(s)/2, (2k + 1)z(s) - h(s) \right\}.$$ 

Note that the right hand side is equal to

$$\min\{\tilde{x}(s_\alpha), l(s_\alpha)/2, (2k + 1)z(s_\alpha) - h(s_\alpha)\},$$

as each of these is decreasing with $s$. To ensure that a.a.s. $t_\alpha < T_D$, we set

$$\epsilon_\alpha = \frac{1}{2} \min\{\tilde{x}(s_\alpha), \bar{\omega}_a - s_\alpha, (2k + 1)z(s_\alpha) - h(s_\alpha)\},$$

which is clearly positive, and set $D = D(\epsilon_\alpha)$.

Thus, by (6.23) and (6.24), for any $\xi$ there is an appropriate $\alpha = O(\xi)$ and $\omega_b = \epsilon_\alpha$, such that for some $t < T_f - \omega_b n$,

$$\tilde{x}((t-t_0)/n) > \alpha z((t-t_0)/n) > 0,$$

and a.a.s.

$$\tilde{X}(t) > \xi Z(t) + \epsilon_c n,$$

for some $\epsilon_c > 0$.

By (6.17), this in turn implies a.a.s. $X(t) > \xi Z(t) + \epsilon_c n$, as required.  

123
6.4 Phase argument

In this section we show that if $T_l$ exists, then $T_g \leq T_l$ exists, which, in combination with Lemmas 6.15 and 6.17, completes the proof that if $X(0) > Z(0) + \epsilon n$ then a.a.s. $X/Z$ will become arbitrarily large significantly before time $T_f$.

**Lemma 6.18** Assuming that $X(0) > Z(0) + \epsilon n$ for some $\epsilon > 0$, a.a.s. if, for some sufficiently small $\epsilon_b > 0$ and $\omega > 0$,

$$T_l(\epsilon_b) < T_f - \omega n,$$

then there exists constant $\epsilon_a > 0$ such that

$$T_g(\epsilon_a) \leq T_l(\epsilon_b).$$

The proof relies on some further analysis of the nature of $\Omega_{\text{ALoad}(k)}$. In particular, the process can be divided into distinct phases, and information about the values of the variables describing the process can be deduced for each phase.

We continue to use the functions $f_a(\lambda)$ and $\psi_l(\lambda)$ defined by (3.5) and (3.6).

6.4.1 Phases of the process

**ALoad**(k) selects an available vertex of highest priority (that is, one for which the priority function returns the smallest value) for processing in each step. For each $s$, $1 \leq s \leq k+1$, the first time when a vertex of priority $s$ is processed can be thought of as the start of a new phase of the process. Or, at any time $t$, given the history, $\mathcal{H}_t$, $\Omega_{\text{ALoad}(k)}$ can be identified as being in a certain phase based on the lowest priority of vertex that has been processed before time $t$.

Either way, by conditioning on the process being in a given phase, it’s possible to deduce information about the values of certain random variables in that phase. In this section we
formalise this idea of phases for $\Omega_{\text{ALoad}(k)}$. Properties of certain variables in each phase are shown in Lemma 6.21.

The following times $T_i$ are each defined as the time when a certain event first occurs. These events are in terms of the values of $Y$, so it is understood that the random variables $X_w$ and $T_w$ defined below are dependent on $Y$, although an argument is omitted. The $\phi_w$ are similar to $\phi_l$: Each $\phi_w(Y(t))$ is the expected number of priority $w$ vertices newly appearing in step $t$, plus the expected number of points newly appearing in priority 1 vertices in step $t$.

**Definition 6.19** Let $X_1 := X_l$ and for $2 \leq w \leq k$, let

$$X_w := \sum_{j=0}^{w-2} Y_{k+w-1,j}.$$  

So, for $w \geq 2$, $X_w(t)$ is the number vertices with priority $w$ at time $t$.

Let $T_0 := 0$ and for $1 \leq w \leq k$ let

$$T_w := \min \left\{ t : \sum_{u=1}^{w} X_u(t) = 0 \right\}.$$  

Note that $T_1 = 0$, $T_u \leq T_{u+1}$ for each $u$, and $T_k = T_h$.

Let $\phi_1(Y) := \phi_l(Y)$ and for $2 \leq w < k$ let

$$\phi_w(Y) := \phi_l(Y) + \frac{1}{L} \sum_{j=0}^{w-2} (k + w - 2j) Y_{k+w,j},$$  

and at time $t$, define $\phi(Y(t))$ to be

$$\phi(Y(t)) := \sum_{w=1}^{k-1} \phi_w(Y(t)) I_{T_w \leq t < T_{w+1}}.$$  

Finally, for $\epsilon_b > 0$ let $T_c(\epsilon_b)$ be

$$T_c(\epsilon_b) := \min\{ t : \phi(Y) > 1 - \epsilon_b \}.$$
Fix constant $\epsilon_b > 0$ for use with the following definition and lemma. We shall use the same $\epsilon_b$ for $T_l(\epsilon_b)$ and $T_e(\epsilon_b)$ and we write $T_l$ for $T_l(\epsilon_b)$ and $T_e$ for $T_e(\epsilon_b)$.

Each $T_l$ determines the start of each new phase, as specified in the next definition, with $T_e$ determining the start of a special and final stage called the explosion phase. The $\phi_i$ are functions whose behaviour determines $T_e$ and hence indicates whether or not the explosion phase starts in phase $i$.

**Definition 6.20** At each time $t$, $\Omega_{\text{ALoad}(k)}$ is said to be in a certain phase according to the following:

- Phase $w := \{t : t < T_e \text{ and } T_w \leq t < T_{w+1}\}$ for $1 \leq w < k$,
- Phase $k := \{t : t < T_e \text{ and } T_k \leq t \leq T_f\}$,
- Explosion phase $:= \{t : T_e \leq t \leq T_f\}$.

Note that in phase $w$, only vertices of priority $w + 1$ or higher are processed. $T_{w+1}$ is the first time that a vertex of priority $w + 2$ is processed. By definition the phase will increment until either $\text{ALoad}(k)$ stops or the explosion phase starts. Each of the $T_w$ and $T_e$ are random variables determined by the history, $H_t$, up to time $t$, and so $H_t$ determines what phase the process is in at time $t$.

Recall that $\rho$ is the density of the $(k + 1)$-core with which $\hat{B}(0)$ is associated, and that from Lemma 6.6

$$\rho = 2k - \frac{1}{n} (X(0) - Z(0)).$$

Recall also the definitions of $H$ and $W$ from the start of Section 6.1. $H(t)$ is a random variable for the number of points, at time $t$, in vertex $n + 1$ of $\hat{B}(t)$ and $W(t)$ is a random variable for the number of vertices of $\hat{B}(t)$ for which the allocation has not yet been generated.

The next lemma gives a description of the behaviour of some of the random variables determined on $\Omega_{\text{ALoad}(k)}$.  

126
Lemma 6.21  Fix $\omega > 0$ and $t < T_f - \omega n$. Condition on $\Omega_{ALoad(t)}$ being in phase $w$ at time $t$. Then the following hold:

(i) $Y_{d,j}(t) = 0$ for $(d, j)$ such that $\text{Pri}_k(d, j) > w + 2$ and $j > 0$.

(ii) A.a.s., for $1 \leq u \leq w$, $\mathbf{E}(X_u(t))$ is at most $O(\log n)$.

(iii) For $w \leq l \leq k$ define $H_l$ and $W_l$ to be

$$H_l(t) := H(t) + \sum_{d=k+l+1}^{2k} dY_{d,0}(t),$$

$$W_l(t) := W(t) + \sum_{d=k+l+1}^{2k} Y_{d,0}(t),$$

with $H_k(t) := H(t)$ and $W_k(t) := W(t)$. A.a.s.

$$H_l(t) = \frac{ne^{\mu_0 - \mu}}{f_{k+1}(\mu_0)} f_{k+l}(\mu) + o(n)$$

and

$$W_l(t) = \frac{ne^{\mu_0 - \mu}}{f_{k+1}(\mu_0)} f_{k+l+1}(\mu) + o(n),$$

where $\mu = \mu(t/n)$, $\mu_0$ is the solution of $\psi_{k+1}(\mu) = \rho$ and $\mu(s)$ is the positive root of $\mu^2(s) = \mu_0^2(\rho - 2s)/\rho$.

Proof. Part (i) follows from the definition of phase. Let $\alpha = (d, j)$ with $\text{Pri}_k(d, j) > w + 2$ and $j > 0$. At $t = 0$ no vertex contains in-points so $Y_{\alpha}(0) = 0$. In phase $w$, the number of vertices of priority $w + 1$ is positive, so only points in vertices of priority $w + 1$ or less can become in-points. When a point in a vertex, $i$ becomes an in-point then the priority of $i$ increases by one (or stays the same if $i$ has priority 1). Hence, all vertices containing in-points have priority $w + 2$ or less, and $Y_{\alpha}(t) = 0$.

For part (ii) we can use the same kind of branching process argument as in the proof of Lemma 6.14, but with $X_l$ replaced by $X_p := X_w + X_l$, and $T'_l$ replaced by $T'_e := \min\{T_e, T_f - \omega n\}$. Note that in analogy with (6.8),

$$\mathbf{E}(\Delta X_p) = -I_{X_p > 0} - \frac{X_l + \sum_{j=0}^{w-2} (k + w - 1 - 2j)Y_{k+w-1,j}}{L} + \phi_w(Y),$$
so \( \phi_w(Y(t)) \) is the expected increase in \( X_p \) in step \( t \). We use \( t \) in the following to mean a general time \( t \), not the fixed \( t \) in the statement of the lemma.

With \( i_2 \) as in Lemma 6.8, define \( D_{d,j}(t) \) to be the event
\[
D_{d,j}(t) := [d^k(i_2) = d \land d^-(i_2) = j \text{ at time } t]
\]
and define \( V(t) \) to be
\[
V(t) := \sum_{r=1}^{k} r I_{D_{2k-r+1,k-r}(t)} + \sum_{j=0}^{w-2} I_{D_{k+w,j}(t)}.
\]
If \( t < T'_e \), \( V(t) \) is the number of priority \( w \) vertices and points in priority 1 vertices newly appearing in step \( t \). Otherwise \( V(t) \) is zero. Note that \( \mathbb{E}(V(t)) = \phi_w(Y(t)) I_{t < T'_e} \).

Fix \( s \) with \( T_w \leq s < T'_e \) and condition on \( X_p(s) = 0 \). Define \( Z_s \) to be the size of the branching process starting at time \( s \) and ending at time \( \bar{s} > s \) if \( X_p(\bar{s}) I_{\bar{s} < T'_e} = 0 \), and with \( V(t) \) children at each step \( s \leq t < \bar{s} \). Now,
\[
P(V(t) = r) = \begin{cases} 
\sum_{i=1}^{k} P_{i} r^i & r = 1 \\
\sum_{j=0}^{w-2} P(D_{k+w,j}(t)) & r = 2, \ldots, k \\
0 & \text{otherwise}.
\end{cases}
\]

Following the same argument as for Lemma 6.14, for \( r = 1, \ldots, k \) and constant \( \delta < \epsilon_b \) define \( P_r \) to be the value of \( P(V(s) = r) \) divided by \( 1 - \delta \), and define \( \tilde{V} \) to be the random variable with distribution
\[
P(\tilde{V} = r) = \begin{cases} 
P_r & r = 1, \ldots, k \\
1 - \sum_{i=1}^{k} P_i & r = 0 \\
0 & \text{otherwise}.
\end{cases}
\]

Then for \( n \) sufficiently large, and for the first \( \lfloor \log^2 n \rfloor \) steps of the process (or the entire process if \( \bar{s} < \lfloor \log^2 n \rfloor \)), \( V(t) \) is stochastically dominated by \( \tilde{V} \).

\( \tilde{V} \) is bounded above by a constant, so has an exponentially small tail, and
\[
\mathbb{E}(\tilde{V}) = \sum_{r=1}^{k} r P_r
\]
\[
= \phi_w(Y(s)) / (1 - \delta)
\]
\[
< (1 - \epsilon_b) / (1 - \delta) < 1.
\]
Define $\tilde{Z}$ to be the size of the branching process with $\tilde{V}$ children at each step. Hence, applying Theorem 1.4, $\tilde{Z}$ has an exponentially small tail. Now, $\tilde{Z}$ stochastically dominates $Z_s$ for $t < \left\lfloor \log^2 n \right\rfloor$, so for $C$ and $n$ sufficiently large,

$$P(Z_s \geq C \log n) \leq P(\tilde{Z} \geq C \log n) = O\left(\frac{1}{n}\right),$$

and it follows from this that

$$E(Z_s) = O(\log n).$$

For $s \leq t < \bar{s}$, $X_p(t)$ is the number of free points in the branching process at time $t$ and so $X_p(t)I_{t < T'_e} \leq Z_s$. This holds for a branching process starting at any time in phase $w$, so

$$\max_{T_w \leq t < T_{w+1}} (E(X_p(t)I_{t < T'_e})) = O(\log n).$$

This implies that for the fixed $t$ in the statement of the lemma and for $u = 1$ and $u = w$, $E(X_u(t))$ is at most $O(\log n)$. If $w > 2$, note that the positive part of $E(\Delta X_{w-1})$ is $\phi_{w-1} - \phi_l$ and

$$\phi_{w-1} - \phi_l = \frac{1}{L} \sum_{j=0}^{w-3} (k + w - 1 - 2j)Y_{k+w-1,j}$$

$$= \frac{1}{L} O(\log n)$$

$$< 1 - \epsilon_b,$$

for $n$ sufficiently large and $t < T'_e$. Hence we may repeat the same branching process argument to show that $E(X_{w-1}(t))$ is at most $O(\log n)$ and by induction on $u$, $E(X_u(t))$ is at most $O(\log n)$ for $1 < u < w$.

For part (iii) we analyse the behaviour of $H_l$ and $W_l$ using differential equations and applying Theorem 1.2. This is almost identical to the analysis of $S(t)$ and $\nu(t)$ in the proof of Lemma 5.1, except that the initial conditions are different, and we choose a different domain, $D$, where the solutions are valid. The following is conditioned on $\Omega_{\text{ALoad}(t)}$ being in phase $w$ at time $t$.

The values of $H_l(0)$ and $W_l(0)$ are as follows. Recall that $\hat{B}(0)$ is the partial pre-allocation corresponding to a $(k+1)$-core, $M_0$, with the distribution of $\mathcal{M}_{k+1}(0, n, \frac{\mu}{T'})$. Let $D =$
Let \((D_1, \ldots, D_n)\) be the degree sequence of \(M_0\). By Lemma 3.6

\[
P(D_1 = i) = p_{k+1,0,i}(1 + o(1)) = \frac{\mu_i}{f_{k+1}(\mu_0)i!} + o(1).
\]

So, with \(Y_{i,0}(0)\) the number of vertices of load-degree \(i\) and in-degree zero in \(\hat{B}(0)\),

\[
E(Y_{i,0}(0)) = \frac{n\mu_i}{f_{k+1}(\mu_0)i!} + o(n),
\]

and by the second moment method, a.a.s.

\[
Y_{i,0}(0) = \frac{n\mu_i}{f_{k+1}(\mu_0)i!} + o(n).
\]

Now

\[
H_l(0) = \rho n - \sum_{i=k+1}^{k+l} iY_{i,0}(0),
\]

\[
W_l(0) = n - \sum_{i=k+1}^{k+l} Y_{i,0}(0).
\]

Hence, a.a.s.

\[
H_l(0) = \rho n - \sum_{i=k+1}^{k+l} \frac{i n\mu_i}{f_{k+1}(\mu_0)i!} + o(n)
= n\psi_{k+1}(\mu_0) - \frac{n\mu_0}{f_{k+1}(\mu_0)} \sum_{i=k+1}^{k+l} \frac{\mu_i^{i-1}}{(i-1)!} + o(n)
= \frac{n\mu_0 f_k(\mu_0)}{f_{k+1}(\mu_0)} - \frac{n\mu_0}{f_{k+1}(\mu_0)}(f_k(\mu_0) - f_{k+l}(\mu_0)) + o(n)
= \frac{n\mu_0 f_{k+l+1}(\mu_0)}{f_{k+1}(\mu_0)} + o(n),
\]

and

\[
W_l(0) = n - \sum_{i=k+1}^{k+l} \frac{n\mu_i}{f_{k+1}(\mu_0)i!} + o(n)
= n - \frac{n}{f_{k+1}(\mu_0)}(f_{k+1}(\mu_0) - f_{k+l+1}(\mu_0)) + o(n)
= \frac{nf_{k+l+1}(\mu_0)}{f_{k+1}(\mu_0)} + o(n).
\]
So to obtain an a.a.s. result it suffices to condition on \( \hat{B}(0) \) with
\[
H_l(0) \sim \frac{n\mu_0 f_{k+1}(\mu_0)}{f_{k+1}(\mu_0)}
\]
and
\[
W_l(0) \sim \frac{n f_{k+1}(\mu_0)}{f_{k+1}(\mu_0)}.
\]
Before time \( T_{w+1} \), the highest priority of vertex that has been processed is \( w + 1 \). In other words, only vertices of load-degree \( k + w \) or less have been processed. This means that in phase \( w \), the effect of ALoad(\( k \)) on vertices of in-degree zero and load-degree greater than \( k + w \), including all those counted by \( H_l \) and \( W_l \), is the same as the effect of ACore(\( k + w + 1 \)), except that points become out-points, rather than being deleted.

Consequently, equations for the expected changes in \( H_l \) and \( W_l \) are exactly the same as the equations (5.9) for \( S \) and (5.10) for \( \nu \), as derived in the proof of Lemma 5.1, but with \( k \) replaced by \( k + l + 1 \), and \( \lambda \) defined to be the solution of \( \psi_{k+l+1}(\lambda) = H_l/W_l \):
\[
E(\Delta H_l \mid H_t) = \frac{-H_l - (k + l)\lambda W_l}{L} + O(1/L)
\]
\[
E(\Delta W_l \mid H_t) = \frac{-H_l + \lambda W_l}{L} + O(1/L).
\]
These are valid for \( t < T_{w+1} \) and for \( H_l - (k + l + 1)W_l \to \infty \) (in analogy with (5.7)), and we may substitute \( L(t) = \rho n - 2t \). For \( \epsilon > 0 \) let
\[
D(\epsilon) := \{(x, z_1, z_2) : z_1 - (k + l + 1)z_2 > \epsilon, -\epsilon < x < \rho/2 - \epsilon, z_1 < \rho, z_2 > -\epsilon/(2(k + l + 1))\}.
\]
The last two constraints ensure that \( D \) is bounded. The first ensures that positive \( \lambda \) exists and the second ensures that \( L > 2\epsilon n \) so \( O(1/L) = o(1) \). We use the same system of equations (5.12), (5.13) and,
\[
\frac{z_1}{z_2} = \frac{\mu f_{k+1}(\mu)}{f_{k+l+1}(\mu)},
\]
(as in (5.14)), with \( \rho \) in place of \( c \) and \( k + l + 1 \) in place of \( k \). By the same arguments used in Section 5.3, hypotheses (i) and (ii) of Theorem 1.2 are satisfied for \( t < \min\{T_D, T_{w+1}\} \) and (iii) is satisfied within \( D \). So applying Theorem 1.2, a.a.s.
\[
H_l(t) = nz_1(t/n) + o(n)
\]
\[
W_l(t) = nz_2(t/n) + o(n),
\]
for \( t \leq \min\{T_D, T_{w+1}, \sigma n\} \), where \( \sigma \) is the supremum of those \( x \) for which these solutions remain in \( D \), and \( z_1 \) and \( z_2 \) are the solutions to the equations with initial conditions

\[
\begin{align*}
  z_1(0) &= \frac{\mu_0 f_{k+h}(\mu_0)}{f_{k+1}(\mu_0)} \\
  z_2(0) &= \frac{f_{k+h+1}(\mu_0)}{f_{k+1}(\mu_0)}.
\end{align*}
\]

Solving the equations yields

\[
\frac{\mu^2}{\rho - 2x} \quad \text{is constant, and} \quad \frac{z_2}{e^{-\mu} f_{k+l+1}(\mu)} \quad \text{is constant}
\]

in analogy with (5.17) and (5.18), and substituting in the initial conditions and using (6.25) gives

\[
\begin{align*}
  \mu^2 &= \mu_0^2 (\rho - 2x)/\rho \\
  z_2 &= e^{\mu_0 - \mu} f_{k+l+1}(\mu) \\
  z_1 &= e^{\mu_0 - \mu} f_{k+1}(\mu).
\end{align*}
\]

This gives the desired approximations for \( H_l(t) \) and \( W_l(t) \). It remains to be shown that these are valid in phase \( w \) for all \( t < T_f - \omega n \). To do this we show that for \( \epsilon \) sufficiently small, a.a.s. \( T_f - \omega n < \min\{T_D, \sigma n\} \).

The solution cannot reach a boundary of \( D \) determined by the third or fourth constraint, nor determined by \( x = -\epsilon \). Now, \( z_1/z_2 = \psi_{k+l+1}(\mu) \), which by Lemma 3.5 is bounded away from \( k + l + 1 \) for \( \mu \) bounded away from zero. Hence, the first constraint of \( D \) is violated when \( \mu = \mu' \) for some \( \mu' = O(\epsilon) \). By (6.26), \( \mu \) becomes small as \( x \) approaches \( \frac{\rho}{2} \), so one of the constraints will be violated first at \( x = \frac{\rho}{2} - \epsilon' \) for some \( \epsilon' = O(\epsilon) \).

Recall that \( T_f := m = \frac{m}{2} \) so \( t < T_f - \omega n \) is equivalent to \( \frac{1}{2} < \frac{\rho}{2} - \omega \). So, with \( x = \frac{\rho}{4} \), by choosing \( \epsilon \) sufficiently small so that \( \epsilon' < \omega/2 \) we can ensure that for \( t < T_f - \omega n \) neither of the first two constraints are violated and the solution stays within \( D = D(\epsilon) \). A.a.s. \( T_D \) is close to \( \sigma n \), so for \( \epsilon \) sufficiently small, a.a.s. \( T_f - \omega n < \min\{T_D, \sigma n\} \) as required.
6.4.2 Some useful functions

In this section we prove several lemmas regarding properties of certain functions derived from the hypothesis of Lemma 6.18. The proof of Lemma 6.18 shall follow directly from these lemmas.

In the following, we assume that the hypothesis of Lemma 6.18 holds. That is, $X(0) > Z(0) + \epsilon n$ and there exist $\epsilon_b, \omega > 0$ such that $T_l(\epsilon_b) < T_f - \omega n$. Note that $\phi_l(Y) \leq \phi(Y)$ by definition, so $\phi_l(Y) > 1 - \epsilon_b$ implies $t \geq T_e(\epsilon_b)$. Hence $T_e(\epsilon_b) \leq T_l(\epsilon_b)$, and $T_l(\epsilon_b) < T_f - \omega n$ implies $T_e(\epsilon_b) < T_f - \omega n$.

So to prove Lemma 6.18 it is sufficient to show that there exists $\epsilon_a > 0$ such that

$$T_g(\epsilon_a) \leq T_e(\epsilon_b) \text{ a.a.s.}$$

Fix $\epsilon_b$ and $\omega$ satisfying the hypothesis of Lemma 6.18. In the following we write $T_l$ and $T_e$ for $T_l(\epsilon_b)$ and $T_e(\epsilon_b)$.

Consider the process at time $T_e$ and condition on $\Omega_{\text{Load}(k)}$ being in phase $w$, so that $\phi_w(Y) > 1 - \epsilon_b$ at time $T_e$. Using Lemma 6.21 parts (i) and (ii), we deduce some information about the values of the variables at time $T_e$. A.a.s.

$$L = H_w + \sum_{j=0}^{w-1} (k + w - 2j)Y_{k+w,j}$$
$$+ \sum_{j=1}^{w} (k + w + 1 - 2j)Y_{k+w+1,j} + O(\log n),$$

$$X = \sum_{j=0}^{w-1} (k - w)Y_{k+w,j} + \sum_{j=1}^{w} (k - w - 1)Y_{k+w+1,j}$$
$$+ \sum_{d=k+w+1}^{2k-1} (2k - d)Y_{d,0} + O(\log n).$$
So \( \phi_w(Y) > 1 - \epsilon_b \) is a.a.s. equivalent to

\[
L(1 - \epsilon_b) \leq L\phi_w(Y) = (k - w)(k - w + 1)Y_{k+w+1,w} + (k - w + 1)(k - w + 2)Y_{k+w,w-1} + \sum_{j=0}^{w-2}(k + w - 2j)Y_{k+w,j} + O(\log n),
\]

which, assuming \( \epsilon_b \) is sufficiently small and \( n \) is sufficiently large, implies a.a.s.

\[
H_w + \sum_{j=1}^{w-1}(k + w + 1 - 2j)Y_{k+w+1,j} < (k - w - 1)(k - w + 1)Y_{k+w+1,w} + (k - w)(k - w + 2)Y_{k+w,w-1}.
\]

This implies

\[
H_w < (k - w + 2)\left(X - \sum_{d=k+w+1}^{2k-1}(2k - d)Y_{d,0}\right),
\]

which is equivalent to

\[
X > \frac{H_w}{(k - w + 2)} + \sum_{d=k+w+1}^{2k} (2k - d)Y_{d,0} = \frac{H_w}{(k - w + 2)} + 2k(W_w - W) - (H_w - H) = Z + 2kW_w - H_w \frac{(k - w + 1)}{(k - w + 2)}.
\]

So

\[
X - Z > 2kW_w - H_w \frac{(k - w + 1)}{(k - w + 2)} \text{ a.a.s. at time } T_e. \tag{6.27}
\]

We will use (6.27) later to deduce some conditions on \( g(X,Z,H) \) and \( T_g \).

First, note that \( X - Z \leq (2k - \rho)n \) by Lemmas 6.6 and 6.8, so

\[
(2k - \rho)n > 2kW_w - H_w \frac{(k - w + 1)}{(k - w + 2)} \text{ a.a.s. at time } T_e. \tag{6.28}
\]

Given any \( t \), we will be able to use (6.28) to check if \( t \leq T_e \) a.a.s., and this will help us determine if \( T_g \leq T_e \) a.a.s. First we will rewrite it as an a.a.s. equivalent relation, in terms of \( \mu \) and the functions \( f_r(\mu) \).
By Lemma 6.21, part (iii), a.a.s. at time $T_e$

\[
2kW_w - H_w \frac{(k-w+1)}{(k-w+2)} = 2k \frac{ne^{\mu_0-\mu}}{f_{k+1}(\mu_0)} f_{k+w+1}(\mu) - \frac{ne^{\mu_0-\mu}}{f_{k+1}(\mu_0)} \mu f_{k+w}(\mu) \frac{(k-w+1)}{(k-w+2)} + o(n)
\]

\[
= \frac{ne^{\mu_0-\mu}}{f_{k+1}(\mu_0)} \left( 2k f_{k+w+1}(\mu) - \mu f_{k+w}(\mu) \frac{(k-w+1)}{(k-w+2)} + o(1) \right),
\]

where $\mu_0$ and $\mu = \mu(t/n)$ are as defined in Lemma 6.21. In the following it is always assumed that $\mu$ is positive.

Note that by Lemma 3.5 part (iii), and since $\psi_{k+1}(\mu_0) = \rho < 2k$, we have

\[
\mu_0 < \rho < 2k.
\]

Define $F(k, w, \mu)$ to be

\[
F(k, w, \mu) := e^{-\mu} \left( 2k f_{k+w+1}(\mu) - \mu f_{k+w}(\mu) \frac{(k-w+1)}{(k-w+2)} \right),
\]

so that the right hand side of (6.28) is a.a.s. close to a constant times $n$ times $F(k, w, \mu)$.

For the left hand side of (6.28), define $F_0(k, \mu)$ to be

\[
F_0(k, \mu) := e^{-\mu} (2k f_{k+1}(\mu) - \mu f_k(\mu)).
\]

It’s easy to see that $F_0(k, \mu) = 0$ has exactly two non-negative solutions at $\mu = 0$ and $\mu = \mu_{2k}$, where $\mu_{2k}$ is the positive solution to $\psi_{k+1}(\mu) = 2k$, and that $F_0(k, \mu)$ is positive for all $0 < \mu < \mu_{2k}$. Also $\mu_0 < \mu_{2k} < 2k$.

By the definition of $\mu_0$, we have $\rho = \mu_0 f_k(\mu_0)/f_{k+1}(\mu_0)$ and

\[
2k - \rho = 2k - \frac{\mu_0 f_k(\mu_0)}{f_{k+1}(\mu_0)} = \frac{e^{\mu_0}}{f_{k+1}(\mu_0)} F_0(k, \mu_0).
\]

So (6.28) implies that

\[
F_0(k, \mu_0) > F(k, w, \mu) \text{ a.a.s. at time } T_e.
\]
Define \( \tau_e = \tau_e(k,w) \) to be
\[
\tau_e := \max\{\tau, 0 < \tau \leq \mu_0 : F(k,w,\tau) = F_0(k,\mu_0)\}.
\]

If no such \( \tau \) exists, set \( \tau_e := \mu_0 \). Then a.a.s. \( \mu(T_e/n) < \tau_e \) (it is shown in the following that \( F(k,w,\mu) > F_0(k,\mu_0) \) for \( \tau_e < \mu \leq \mu_0 \)).

As \( \mu \) is decreasing with \( t \), this effectively gives an a.a.s. lower bound for \( T_e \). Lemma 6.23 provides a test to check if a given \( \mu \) is greater than \( \tau_e \), which shall be used to show that \( T_g < T_e \) a.a.s. in the case of small \( k \). First we show a simple lemma which shall be used in the proof of Lemma 6.23

**Lemma 6.22** Let \( g : [0, \infty) \to \mathbb{R} \) be a continuous function with all derivatives continuous on \([0, \infty)\). Suppose that \( g(0), g'(0) \leq 0 \) and \( g(x), g'(x) \to \infty \) as \( x \to \infty \). If \( g'(x) = 0 \) has at most 1 positive solution, then \( g(x) = 0 \) has at most 1 positive solution also.

**Proof.** If \( g'(x) = 0 \) has no positive solution, then \( g(x) = 0 \) is monotonically increasing for \( x > 0 \), so clearly \( g(x) = 0 \) has at most 1 positive solution. Assume \( g'(x) = 0 \) has a unique positive solution, \( x_0 \). Then \( g(x_0) \) is the unique turning point of \( g(x) \). It cannot be a maximum as \( g(x) \) tends to infinity. Hence \( g(x) \) is monotonically increasing for \( x > x_0 \) and monotonically increasing or decreasing for \( 0 < x < x_0 \). It’s not possible that \( g(x_0) = g(0) \).

If \( g(x_0) > g(0) \) then \( g(x) \) is monotonically increasing for \( 0 < x < x_0 \) also, so clearly \( g(x) = 0 \) has at most 1 solution. If \( g(x_0) < g(0) \) then \( g(x_0) < 0 \) and \( g(x) \) is monotonically decreasing and negative for \( 0 < x < x_0 \). Hence \( g(x) = 0 \) has no solution in the range \( 0 < x \leq x_0 \), and exactly 1 solution for \( x > x_0 \) since \( g(x) \) tends to infinity. \( \blacksquare \)

**Lemma 6.23** Let \( \mu_1 = \mu_1(t_1/n) \) for some \( t_1 > 0 \) and \( \mu_1 > 0 \). For fixed \( k \) and \( w \in [k-1] \), if
\[
F(k,w,\mu_1) > F_0(k,\mu_1) \quad \text{and} \quad \frac{\partial F_0}{\partial \mu}(k,\mu_1) < 0, \quad (6.31)
\]
then
\[
F(k,w,\mu) > F_0(k,\mu_0)
\]
for all $\mu$, $\mu_1 \leq \mu \leq \mu_0$, and

$$\mu_1 > \tau_e(k, w).$$

If (6.31) holds for each $w \in [k - 1]$ then

$$t_1 < T_e \text{ a.a.s.}$$

**Proof.** Assume $k$ and $w$ are fixed and that $\mu_1$ satisfying the hypothesis of the lemma exists. $F(k, w, \mu) > F_0(k, \mu)$ if and only if

$$2k f_{k+w+1} - \frac{k - w + 1}{k - w + 2} \mu f_{k+w} > 2k f_{k+1} - \mu f_k,$$

which is equivalent to each of the following,

$$0 > 2k \sum_{i=1}^{w} \frac{\mu^{k+i}}{(k+i)!} - \mu \sum_{i=0}^{w-1} \frac{\mu^{k+i}}{(k+i)!} - \frac{\mu f_{k+w}}{k - w + 2}$$

$$\frac{\mu f_{k+w}}{k - w + 2} > \sum_{i=1}^{w} \frac{(2k - (k+i)) \mu^{k+i}}{(k+i)!}$$

$$f_{k+w} > (k - w + 2) \sum_{i=1}^{w} \frac{(k - i) \mu^{k+i-1}}{(k+i)!}$$

$$e^\mu > 1 + \sum_{i=1}^{k+w-1} \frac{\mu^i}{i!} + (k - w + 2) \sum_{i=1}^{w} \frac{(k - i) \mu^{k+i-1}}{(k+i)!}$$

$$0 < e^\mu - 1 - \sum_{i=1}^{k+w-1} \frac{\mu^i}{i!} - (k - w + 2) \sum_{i=k}^{k+w-1} \frac{(2k - i - 1) \mu^i}{(i+1)!}$$

$$=: P(\mu).$$

Note that $P(0) = 0$. We will apply Lemma 6.22 to show that $P(\mu) = 0$ has at most 1 positive solution. Let $J = k + w - 1$. Then (using $0! := 1$)

$$P(\mu) = e^\mu - \sum_{i=0}^{J} A_{i,0} \frac{\mu^i}{i!},$$

where

$$A_{i,0} = \begin{cases} 1 & \text{for } 0 < i < k \\ 1 + \frac{(k-w+2)(2k-i-1)}{i+1} & \text{for } k \leq i \leq J \\ 0 & \text{otherwise,} \end{cases}$$

137
and the $d$th derivative of $P(\mu)$ is

$$P^{(d)}(\mu) = e^\mu - \sum_{i=0}^{J-d} A_{i,d} \frac{\mu^i}{i!},$$

with

$$A_{i,d} = \begin{cases} 1 & \text{for } 0 < i < k-d \\ 1 + \frac{(k-w+2)(2k-i-d-1)}{i+d+1} & \text{for } \max(k-d,0) \leq i \leq J-d \end{cases}$$

All derivatives of $P(\mu)$ are continuous on $[0, \infty)$ and for each $d \leq J$, $P^{(d)}(0) = 1 - A_{0,d} \leq 0$, since each $A_{i,d} \geq 1$. Clearly $P^{(d)}(\mu) \to \infty$ as $\mu \to \infty$ since $e^\mu$ is the dominant term. Now

$$P^{(J)}(\mu) = e^\mu - A_{J,0},$$

which is monotonically increasing, so $P^{(J)}(\mu) = 0$ has 1 positive solution if $A_{J,0} > 1$ and none otherwise. By Lemma 6.22, $P^{(J-1)}(\mu) = 0$ has at most 1 positive solution, and by induction $P^{(d)}(\mu) = 0$ has at most 1 positive solution for each $d < J-1$. Hence $P(\mu) = 0$ has at most 1 positive solution.

Let $\mu_P$ be the maximum solution of $P(\mu) = 0$. Then, since $P(\mu)$ tends to infinity, $P(\mu) < 0$ for all $0 < \mu < \mu_P$ and $P(\mu) > 0$ for all $\mu > \mu_P$. Hence, for $\mu > 0$,

$$F(k, w, \mu) > F_0(k, \mu) \text{ if and only if } \mu > \mu_P.$$ 

So $F(k, w, \mu_1) > F_0(k, \mu_1)$ implies $F(k, w, \mu) > F_0(k, \mu)$ for all $\mu \geq \mu_1$.

Next,

$$\frac{\partial F_0}{\partial \mu} = e^{-\mu}(-2kf_{k+1} + \mu f_k + (2k-1)f_k - \mu f_{k-1})$$

$$= e^{-\mu} \left(2k \frac{\mu^k}{k!} - \mu \frac{\mu^{k-1}}{(k-1)!} - f_k \right)$$

$$= e^{-\mu} \left(k \frac{\mu^k}{k!} - f_k \right).$$

This is zero at $\mu = 0$ and $\mu = \mu_p$, the unique positive solution to $f_k = \frac{\mu^k}{(k-1)!}$. Note that $\frac{\mu^k}{(k-1)!} - f_k$ is positive for $0 < \mu < \mu_p$ and negative for $\mu > \mu_p$. 

138
Hence, \( \frac{\partial F_0}{\partial \mu}(k, \mu_1) < 0 \) implies \( \mu_1 > \mu_p \) and therefore \( F_0(k, \mu) > F_0(k, \mu_0) \) for all \( \mu_1 \leq \mu \leq \mu_0 \). This and the previous conclusion together imply the first statement of the lemma.

The second follows from the first and the definition of \( \tau_e \). The third statement follows from (6.30) and because \( \mu \) is decreasing with \( t \).

Now we examine what can be deduced about the values of the variables if \( T_g \) is not reached. Assume that \( T_g < T_f \) does not exist. Then for all \( t < T_f \),

\[
g(X, Z, H) < 1 + \epsilon_a.
\]

Assuming \( Z > 0 \), this is equivalent to

\[
\frac{X}{Z} < \frac{1 + \epsilon_a}{1 - 2Z/H},
\]

\[
X < Z \left( \frac{1 + \epsilon_a}{1 - 2Z/H} \right) - 1,
\]

\[
X - Z < Z \left( \frac{1 + \epsilon_a}{1 - 2Z/H} \right) - 1
\]

\[
= Z \left( \frac{\epsilon_a + 2Z/H}{1 - 2Z/H} \right)
\]

\[
= \frac{2Z^2 + \epsilon_a HZ}{H - 2Z}.
\]

So, for any \( \epsilon_a > 0 \), there exists \( \bar{\epsilon}_a = O(\epsilon_a) \) such that if for some \( t < T_f \),

\[
X(t) - Z(t) > \frac{2Z(t)^2}{H(t) - 2Z(t)} + \bar{\epsilon}_a,
\]

then \( g(X, Z, H) > 1 + \epsilon_a \) at this time, and hence \( t \geq T_g \).

We will use (6.32) in combination with (6.27) to deduce a condition that we use to show that, conditioning on the explosion phase starting in phase \( w \) and \( T_e < T_f - \omega n \), a.a.s. \( T_g \leq T_e \) exists. First we rewrite the right hand side of (6.32) in terms of \( \mu \) and the functions \( f_r(\mu) \).

By Lemma 6.21 part (iii), a.a.s.

\[
H(t) = \frac{n e^{\mu_0 - \mu}}{f_{k+1}(\mu_0)} \mu f_{2k}(\mu) + o(n),
\]

\[
Z(t) = \frac{n e^{\mu_0 - \mu}}{f_{k+1}(\mu_0)} (\mu f_{2k}(\mu) - 2k f_{2k+1}(\mu)) + o(n).
\]
Hence, for \( t < T_f - \omega n \), a.a.s.

\[
\frac{2Z^2}{H - 2Z} = \frac{ne^{\mu_0 - \mu}}{f_{k+1}(\mu_0)} \left( \frac{2(\mu f_{2k}(\mu) - 2kf_{2k+1}(\mu) + o(1))}{4kf_{2k+1}(\mu) - \mu f_{2k}(\mu)} + o(1) \right).
\]

Define \( D(k, \mu) \) to be

\[
D(k, \mu) := e^{-\mu} \frac{2(\mu f_{2k}(\mu) - 2kf_{2k+1}(\mu))^2}{4kf_{2k+1}(\mu) - \mu f_{2k}(\mu)}.
\]

(6.33)

Let \( \mu_f := \mu \left( \frac{T_f - \omega n}{n} \right) \). Then \( \mu_f > 0 \). Define \( \tau_g = \tau_g(k, w) \) to be

\[
\tau_g := \min \{ \tau \geq \mu_f : F(k, w, \tau) = D(k, \tau) \}.
\]

If no such \( \tau_g \) exists set \( \tau_g = \infty \).

**Lemma 6.24** For fixed \( k \) and \( w \in [k-1] \), and \( \mu_f < \mu_1 \leq 2k \), if

\[
F(k, w, \mu_1) - D(k, \mu_1) > 0,
\]

(6.34)

then there exists \( \delta > 0 \) such that \( F(k, w, \mu) - D(k, \mu) > \delta \) for all \( \mu_f < \mu \leq \mu_1 \), and

\[
\mu_1 < \tau_g(k, w).
\]

**Proof.** Now,

\[
e^\mu (F(k, k-r, \mu) - D(k, \mu)) = 2k f_{2k-r} - \mu f_{2k-r} \left( \frac{r+1}{r+2} \right) - \frac{2(\mu f_{2k} - 2kf_{2k+1})^2}{4kf_{2k+1} - \mu f_{2k}}
\]

\[
= 2k \sum_{2k-r+1}^{2k} \frac{\mu^i}{i!} + 2k f_{2k+1} - \mu f_{2k} - \mu \sum_{2k-r}^{2k-1} \frac{\mu^i}{i!} + \mu f_{2k-r} \frac{r+2}{r+2} - \frac{2(2kf_{2k+1} - \mu f_{2k})^2}{4kf_{2k+1} - \mu f_{2k}}
\]

\[
= \sum_{2k-r+1}^{2k} \frac{(2k - i) \mu^i}{i!} + \frac{\mu f_{2k-r}}{r+2} + (2k f_{2k+1} - \mu f_{2k}) \left( 1 - \frac{2(2kf_{2k+1} - \mu f_{2k})}{4kf_{2k+1} - \mu f_{2k}} \right)
\]

\[
= \sum_{j=1}^{r-1} \frac{j \mu^{2k-j}}{(2k - j)!} + \frac{\mu f_{2k}}{r+2} + \frac{\mu}{r+2} \sum_{j=1}^{r} \frac{\mu^{2k-j}}{(2k - j)!} + \frac{(2kf_{2k+1} - \mu f_{2k}) \mu f_{2k}}{4kf_{2k+1} - \mu f_{2k}},
\]

(6.35)
Let
\[ R(k, r, \mu) := \frac{(2k)!e^\mu}{\mu^{2k}(\mu)} (F(k, k - r, \mu) - D(k, \mu)). \]

Then for \( \mu > \mu_f \), \( F(k, k - r, \mu) - D(k, \mu) > 0 \) if and only if \( R(k, r, \mu) > 0 \), and
\[
R(k, r, \mu) = \sum_{j=1}^{r-1} j\frac{[2k]!\mu^{2k}}{\mu^{j+1} f_{2k}} + \frac{(2k)!}{(r + 2) f_{2k}} \sum_{j=1}^{r} \frac{[2k]_j}{\mu^j} + \frac{(2k)!(2kf_{2k+1} - \mu f_{2k})}{(4kf_{2k+1} - \mu f_{2k})}.
\]

Now
\[
\frac{f_{2k}(\mu)}{\mu^{2k}} = \sum_{i \geq 0} \frac{\mu^i}{(2k + i)!}.
\]

This increases with \( \mu \), and in the limit as \( \mu \) approaches zero is \( 1/(2k)! \). It follows that the first and third terms in \( R(k, r, \mu) \) are decreasing from infinity as \( \mu \) increases from zero, whilst the second term is constant.

Regarding the last term,
\[
\frac{2kf_{2k+1} - \mu f_{2k}}{4kf_{2k+1} - \mu f_{2k}} = \frac{2k - \psi_{2k+1}(\mu)}{4k - \psi_{2k+1}(\mu)} = 1 - \frac{2k}{4k - \psi_{2k+1}(\mu)} \rightarrow -1 \quad \text{as} \quad \mu \rightarrow 0^+,
\]
by Lemma 3.5, part (ii). By part (v) of the same lemma, \( \psi_{2k+1}(\mu) \) increases with \( \mu \), so the last term of \( R(k, r, \mu) \) is negative and decreasing with \( \mu \) provided \( \psi_{2k+1}(\mu) < 4k \), which is the case for \( \mu \leq 2k \) by Lemma 3.5, part (iv). Note also that \( R(k, r, \mu) \) is continuous in this range of \( \mu \).

Hence \( R(k, r, \mu) \) decreases with \( \mu \) and, as the first and third terms go to infinity as \( \mu \) goes to zero, \( R(k, r, \mu) \) does also. This implies that \( F(k, k - r, \mu) - D(k, \mu) \) is also continuous and positive for all \( 0 < \mu \leq \mu_1 \), from which the second statement of the lemma follows by the definition of \( \tau_g \).

Let
\[
\delta = \frac{1}{2} \min_{\mu_f \leq \mu \leq \mu_1} (F(k, w, \mu) - D(k, \mu)).
\]
Because \( F(k, k - r, \mu) - D(k, \mu) \) is continuous and positive for \( \mu_f \leq \mu \leq \mu_1 \), \( \delta > 0 \) exists and the first statement of the lemma follows. \( \blacksquare \)
Lemma 6.25 A.a.s. if, for fixed $k$ and for each $w \in [k-1]$,

$$\tau_g(k, w) > \tau_e(k, w),$$

then there exists $\epsilon_a > 0$ such that

$$T_e > T_g.$$

Proof. If $\tau_g(k, w) > \tau_e(k, w)$ then $F(k, w, \tau_e) > D(k, \tau_e)$ and by Lemma 6.24 there exists $\delta > 0$ such that $F(k, w, \mu) > D(k, \mu) + \delta$ for all $\mu_f < \mu \leq \tau_e$. In particular, as $T_e < T_f - \omega_n$, this range of $\mu$ includes $\mu(T_e/n)$ a.a.s.

So by (6.27) and (6.32) and the definitions of $F(k, w, \mu)$ and $D(k, \mu)$, this implies $T_e > T_g$ a.a.s. □

6.4.3 Proof of Lemma 6.18

The small $k$ and large $k$ cases are treated separately by the next two lemmas.

Lemma 6.26 For $k = 2, \ldots, 6$ and each $w \in [k-1]$,

$$\tau_g(k, w) > \tau_e(k, w).$$

Proof. Table 6.26 gives $\frac{\partial F_0}{\partial \mu}$, $F_0$, $F$ and $D$ for each $k$ and $w$, evaluated at a specific value of $\mu$.

$F_0'$ denotes $\frac{\partial F_0}{\partial \mu}$, $F_{w=1}$ denotes $F(k, i, \mu_1)$, and $F_m$ denotes $\min\{F(k, i, \mu_1), \ i \in [k-1]\}$. All functions are evaluated at $\mu = \mu_1$. Firstly, observe that for each $k$ and $w$, $\frac{\partial F_0}{\partial \mu}(k, \mu_1) < 0$ and $F_0(k, \mu_1) < F(k, w, \mu_1)$, so by Lemma 6.23, $\mu_1 > \tau_e$. Secondly, observe that for each $k$ and $w$, and $D(k, \mu_1) < F(k, w, \mu_1)$, so by Lemma 6.24, $\mu_1 < \tau_g$. Hence $\tau_e < \mu_1 < \tau_g$ and the lemma follows. □

Lemma 6.27 For all $k \geq 7$ and each $w \in [k-1]$,

$$F(k, w, 2k) - D(k, 2k) > 0.$$
The proof of this lemma is rather long and can be found in Appendix A.

**Proof of Lemma 6.18.** Lemma 6.26 showed that for $2 \leq k \leq 6$ and each $w \in [k-1]$, $\tau_g(k, w) > \tau_e(k, w)$, whilst Lemma 6.27 showed that for $k \geq 7$ and each $w \in [k-1]$, $F(k, w, 2k) - D(k, 2k) > 0$ and hence by Lemma 6.24 and the definition of $\tau_e$, $\tau_g(k, w) > 2k > \tau_e(k, w)$ for each $w \in [k-1]$.

Hence, by Lemma 6.25, for any fixed $k \geq 2$, $T_e > T_g$ a.a.s. As $T_l \geq T_e$ (see comments at the start of Section 6.4.2), this implies $T_l > T_g$ a.a.s., as required. ■
Chapter 7

The end of the load balancing process

None of the methods used so far can be extended to the end of $\Omega_{\text{ALoad}(k)}$. Further to the discussion of Section 6.1.1, it is sufficient to analyse the process up to time $T_h := \min\{t : X(t) = 0\}$, but as $X$ becomes small, the hypotheses of Theorem 1.2 become invalid and we cannot extrapolate the continuous approximations of Lemma 6.17 to conclude from $X/Z$ becoming arbitrarily large that $Z$ goes to zero before $X$.

Instead, we investigate more directly the process applied to a pseudograph for which $Z = o(X)$. In this chapter we show that the outcome suggested by the conclusion of Lemmas 6.15, 6.17 and 6.18 is correct. In particular, we show the following lemma.

**Lemma 7.1** For any $\delta > 0$, there is a constant $\xi = \xi(\delta)$ such that, for any constants $\epsilon_c, \omega > 0$, a.a.s. if there exists $T < T_f - \omega n$ such that $X(T) > \xi Z(T) + \epsilon_c n$, then $P(Z(T_h) = 0) > 1 - \delta$.

Recall the role of vertex $n + 1$ in the partial pre-allocation $\hat{B}(T)$. Namely, the points in vertex $n + 1$ represent the points in heavy vertices of a pseudograph corresponding to $\hat{B}(T)$. The allocation of these points is yet to be generated by $\text{ALoad}(k)$. 

144
$Z(T_h)$ will be zero if enough of the $H(T)$ points in vertex $n + 1$ of $\hat{B}(T)$ become out-points by time $T_h$. In terms of a pseudograph this is the same as allocating a sufficient number of the edges adjacent to heavy vertices to other vertices. $X(t)$ is the number of places where edges can be allocated, or the number of points which could become in-points without the maximum in-degree exceeding $k$, and $Z(t)$ is the number of edges which need to be allocated away from heavy vertices.

Qualitatively speaking, if $X(t)/Z(t) \to \infty$ it seems that it should be possible to do this, unless a dense subgraph containing heavy vertices manages to survive somehow until $T_h$ (see Lemma 1.1). The likelihood of such a subgraph seems very small, and yet, the proof of Lemma 7.1 is quite involved. Firstly, we fix the degree sequence of the points in vertex $n + 1$ of $\hat{B}(T)$, and replace the partial pre-allocation $\hat{B}(T)$ with a pre-allocation $\hat{A}(T)$. Then we define a colouring algorithm which builds a matching, $M_F$, on some of the points in the allocation as the algorithm $\text{ALoad}(k)$ completes. Finally, we can deduce properties of this matching which imply that $P(Z(T_h) = 0)$ is arbitrarily close to 1 a.a.s.

The definition of $M_F$ involves several steps and rules for colouring points red, blue, white or black. However, behind the rather complicated details is the basic idea of needing to allocate edges away from heavy vertices and towards vertices which can take some load (those which make a positive contribution to $X(t)$). Another way of saying this is that we can identify sources and sinks for load, and $M_F$ is a matching of these sources and sinks. The generation of $M_F$ actually involves identifying directed paths of allocated edges (these paths are not complete until time $T_h$) which go from heavy vertices to vertices that, at some time in the process, have degree less than $2k$. These are paths from the sources to the sinks and can be of length greater than one.

The emerging graph of these paths we call $F_t$, and the pairs of $M_F$ are defined as the pairs of endpoints of the paths of $F_{T_h}$. Each of the $H(T)$ points in heavy vertices at time $T$ is a source, and these are coloured red. There is the potential for $X(T)$ sinks, and $X(T)$ points get coloured blue dynamically. $M_F$ is defined so as to contain all $H(T) + X(T)$ of the red and blue points.
The motivation for defining $M_F$ is that, firstly, any point matched to a blue point in $M_F$ is not free at time $T_h$. We define a graph, $G_R$, from $\hat{A}(T_h)$ and the red-red pairs in $M_F$, and show in Lemma 7.8 that any red point surviving at time $T_h$ is in the 2-core of $G_R$. Secondly, on showing that $M_F$ has the distribution of a uniformly random matching (Lemma 7.6), we deduce that $G_R$ can be generated by a simple matching process, and in Lemma 7.11 we analyse a related branching process to bound the probability that $G_R$ contains a cycle. Hence, we find the desired bound on the probability that $Z(T_h) \neq 0$.

Some fiddly details arise in the precise definition of $M_F$, mainly because the way $\text{ALoad}(k)$ allocates edges is random and so $F_t$ grows in an ad-hoc way. The definition of $M_F$ has to ensure that a set of paths and matching with the desired properties is obtained (there are any number of ways that a set of paths could be identified in a directed pseudograph).

The proof of Lemma 7.1 is in Section 7.2. First, to foreshadow the definition of $M_F$, we describe some ways in which a uniform random matching can be generated.

### 7.1 Generating a random matching

Suppose we have $r$ red points and $b$ blue points. A uniformly random matching of these $r + b$ points can be generated by, at each step, choosing one point, then choosing a second point uniformly at random to match with the first. If we restrict to the first point chosen being blue as long as there are unmatched blue points, then the resulting matching is still uniformly random.

When a point, $p$, gets matched to a point $q$, say that $p$ is the *mate* of $q$ and visa-versa.

Suppose there are also $w$ white points. Consider the following algorithm.

**Algorithm 7.2**

Repeat the following:

- If there are unmatched blue points
  - choose an unmatched blue point, $p$, by any rule
- Else, choose an unmatched red point, $p$, by any rule
Repeat
  choose a point, \( q \), u.a.r. from unmatched points other than \( p \)
  If \( q \) is red or blue
    match \( p \) and \( q \)
  Else
    discard \( q \)
  Until \( p \) has a mate
Until all red and blue points are matched
End

Once again, this will generate a uniformly random matching of the \( r \) red and \( b \) blue points.
Now consider the following algorithm which starts with \( r \) red points and \( b + w \) uncoloured points, \( b \) of which get coloured blue dynamically.

**Algorithm 7.3**

\[
B := 0; \quad U := b + w.
\]

Repeat the following:
  If \( B < b \),
    choose an uncoloured point, \( p \), u.a.r.
    colour \( p \) blue
    \( B \leftarrow B + 1 \), \( U \leftarrow U - 1 \)
  Else, choose an unmatched red point, \( p \), by any rule
Repeat
  choose a point, \( q \), u.a.r. from unmatched points other than \( p \)
  If \( q \) is red
    match \( p \) and \( q \).
  Else, with probability \( \frac{b-B}{U} \)
    colour \( q \) blue.
    match \( p \) and \( q \).
    \( B \leftarrow B + 1 \), \( U \leftarrow U - 1 \)
Else,

discard $q$.

$U \leftarrow U - 1$

Until $p$ has a mate

Until $B = b$ and all red points are matched

End

Note that $B$ is the number of matched blue points and $U$ is the number of uncoloured points. Similarly, let $R$ be the number of matched red points. Then each time $q$ is selected

\[
\begin{align*}
P(q \text{ is red } &= \frac{r - R}{U + r - R}, \\
P(q \text{ is blue } &= \frac{U}{U + r - R} \times \frac{b - B}{U} = \frac{b - B}{U + r - R}, \\
P(q \text{ is discarded } &= \frac{U}{U + r - R} \left(1 - \frac{b - B}{U}\right) = \frac{U - b + B}{U + r - R}, \\
P(q \text{ is kept } &= 1 - \frac{U - b + B}{U + r - R} = \frac{r - R + b - B}{U + r - R}.
\end{align*}
\]

So the probability that $p$ gets matched to a blue point is

\[
P(q \text{ is blue } | q \text{ is kept}) = \frac{P(q \text{ is blue } \land q \text{ is kept})}{P(q \text{ is kept})} = \frac{P(q \text{ is blue })}{P(q \text{ is kept})} = \frac{b - B}{r - R + b - B},
\]

and hence, the probability that $p$ gets matched to a red point is

\[
P(q \text{ is red } | q \text{ is kept}) = \frac{r - R}{r - R + b - B}.
\]

These are exactly the same probabilities as if the matching algorithm was the first simple one mentioned. This, and that $q$ is chosen u.a.r., ensure that the resulting matching is a uniformly random one.

We will define a similar algorithm for building a uniformly random matching on points in an allocation to which $\text{ALoad}(k)$ is applied. The definition will be somewhat more complicated, but the basic principle described here will still apply.
7.2 Random matching argument

Proof of Lemma 7.1, part 1. Fix $\delta > 0$ and let $\xi = K/\delta^\frac{1}{2}$ for some large $K = K(k)$. We will show the lemma for $K$ sufficiently large. Fix $\epsilon_c$, $\omega$ and assume that there exists $T < T_f - \omega n$ such that $X(T) > \xi Z(T) + \epsilon_c n$. We aim to show that $P(Z(T_h) = 0) > 1 - \delta$, a.a.s.

Fix $\epsilon_d > 0$ satisfying $\epsilon_d < \frac{\epsilon_c}{\xi}$. We may assume that $Z(T) > \epsilon_d n$ by the following: Suppose $Z(T) < \epsilon_d n$. Then, assuming $n$ is sufficiently large, there exists $T' < T$ such that $2\epsilon_d n > Z(T') > \epsilon_d n$. Now

$$X(T') \geq X(T)$$
$$> \epsilon_c n$$
$$> 2\xi \epsilon_d n + \frac{\epsilon_c}{2} n$$
$$> \xi Z(T') + \frac{\epsilon_c}{2} n.$$

Hence, with $\epsilon_c$ replaced by $\frac{\epsilon_c}{\frac{\epsilon_c}{\xi}}$, $T'$ satisfies the conditions of the lemma and $Z(T') > \epsilon_d n$. So $T$ can be replaced by $T'$, if required.

By Lemma 6.9, for all $t \geq T_h$, $Z(t) \geq X(t)$. Hence, $T < T_h$. It follows that there are no vertices of degree $2k + 1$ in $\hat{B}(T)$ and $Z(t) = H(t) - 2k W(t)$ for all $T \leq t \leq T_h$. In particular, $Z(T_h) = 0$ if and only if $H(T_h) = 0$.

Let $X_T$, $Z_T$, $H_T$ and $W_T$ be the values of $X$, $H$ and $W$ at time $T$.

Rather than work with $\hat{B}(T)$, let $\hat{A}_T$ be a partially oriented pre-allocation with the same low load-degree sequence as $\hat{B}(T)$, say $(d_T, j_T)$, and with high degree sequence (of vertices of degree greater than $2k$) with the distribution of $\text{Multi}(2k + 1, W_T, H_T)$. Note that $\hat{A}_T$ is one step closer to a pseudograph than $\hat{B}(T)$; the degree sequence is known, though the pairing of points is yet to be generated. The degree sequence of $\hat{A}_T$ has the same distribution as that of a random pseudograph in $\mathcal{M}_{2k+1}(d_T, j_T, W_T, H_T)$, the restricted space of pseudographs associated with $\hat{B}(T)$.

We prove Lemma 7.1 by analysing $\text{ALoad}(k)$ applied to $\hat{A}_T$ (the allocation generation
steps are now redundant as there is no vertex $n+1$ in $\hat{A}_T$) and showing that for a “reasonable” load-degree sequence $(d, j)$, $\text{ALoad}(k)$ applied to $\hat{A}_T = A(d, j)$ will have a successful output a.a.s. What is a reasonable load-degree sequence is made precise in Definition 7.9, and in Lemma 7.10 it is shown that the load-degree sequence of $\hat{A}_T$ is a.a.s. reasonable. In combination, these imply that $\text{ALoad}(k)$ applied to $\hat{B}_T$ will have a successful output a.a.s. We omit a rigorous justification of the validity of replacing $\hat{B}_T$ with $\hat{A}_T$, but note that one could be made using Lemma 4.2 and arguments along the lines of those used to prove Lemma 4.25.

Let $\hat{A}(t)$ be the allocation at time $t$. Now, for $T \leq t \leq T_h$, $W(t)$ and $H(t)$ are interpreted as the number of vertices of degree greater than $2k$, and the number of free points in these vertices, respectively. $Z(t)$ remains as $Z(t) = H(t) - 2kW(t)$.

Let the $H_T$ free points in vertices of degree greater than $2k$ in $\hat{A}_T$ be coloured red. If there are no free red points at time $T_h$ then $H(T_h) = 0$ and $Z(T_h) = 0$.

In the following we will say that a point survives if it is free at time $T_h$. We will show that the probability that at least one red point survives at time $T_h$ tends to zero (which is a sufficient, though not necessary condition), and hence that $Z(T_h) = 0$.

The proof proceeds as follows. Firstly, a random matching, $M_F$, is defined between the red points and some other points in $\hat{A}_T$ defined dynamically during the process. Secondly, we interrupt the proof to show that the distribution of this matching will be uniform at random on the matched red points, and other properties that imply that any surviving red points must be sparse in $\hat{A}(T_h)$. At this stage we will be able to draw the conclusion that there is a small upper bound on the number of red points surviving at time $T_h$. Thirdly, we define a pseudograph, $G_R$, determined by $\hat{A}(T_h)$ and the red-red pairs of $M_F$ and show that any surviving red points are in the 2-core of $G_R$. Then we analyse a branching process that generates $G_R$ and show that a.a.s. if $G_R$ is sparse it is acyclic with high probability. Finally we use these results to conclude the proof of Lemma 7.1.

For the first part of the proof we will derive from $\text{ALoad}(k)$ a set of paths and a matching, $M_F$, of $X_T + H_T$ points, $H_T$ of which will be the red points and $X_T$ of which will be in-
points or out-points at time $T_h$. Because $X_T$ gives a lower bound on the number of points which will be processed from time $T$ until $T_h$, this number of in-points and out-points can be found. Anything matched to an in-point or an out-point in $M_F$ will not have survived. In this way, the number of red points matched to a red point in $M_F$ will give an upper bound for the number of red points which survive.

Note that $X_T + H_T$ must be even, since the total number of free points, $H + \sum_{d \leq 2k} (d - 2j)Y_{d,j}$, is even and, using $\equiv$ to denote equivalence modulo two,

$$X + H \equiv H + \sum_{d \leq 2k} (d - 2j)Y_{d,j} - (X + H)$$

$$\equiv \sum_{d \leq 2k} (d - 2j)Y_{d,j} - \sum_{(d,j) \in B} (2k - d)Y_{d,j} - \sum_{(d,j) \in A} (d - 2j)Y_{d,j}$$

$$\equiv \sum_j (2k - 2j)Y_{2k,j} + \sum_{(d,j) \in B} (d - 2j - 2k + d)Y_{d,j}$$

$$\equiv 0,$$

with the sets $A$ and $B$ as defined on page 100.

For each vertex $i \leq n$ let $d_T(i)$ and $j_T(i)$ denote the load-degree and in-degree of $i$ in $\hat{A}_T$.

Let

$$\bar{b}(i) := \min(2k - d_T(i), d_T(i) - 2j_T(i)).$$

Then

$$X_T = \sum_{i \leq n} \bar{b}(i). \quad (7.1)$$

As $\text{ALoad}(k)$ progresses after time $T$, we define certain points to be coloured blue, and some to be coloured black or white, with the white and black points within each vertex always linked in pairs (by special links, not pairs in the pairing).

After step $t$, let $F_t$ denote the graph whose vertices are points and whose edges are linked pairs and pairs of the pairing (the maximum degree of vertices in $F_t$ is 2). The colouring and linking will be done so that all the paths in $F_t$ have a blue, red or white point at each end. One can think of the white points as free points at the end of a path "waiting" until it reaches either a blue or red point. Black points are exposed and are in the interior of paths.

151
Initially (at time $T$), for each vertex $i \leq n$, if $i$ is a priority 1 vertex then $d_T(i) - j_T(i) \leq k$ and all $d_T(i) - 2j_T(i)$ free points in $i$ (in this case $\bar{b} = d_T(i) - 2j_T(i)$) are coloured blue. Other free points in vertices $1, \ldots, n$ are uncoloured. No points are initially white or black.

At time $T_h$, there will be no white points remaining and all paths in $F_t$ will have a blue or red point at each end. Moreover, there will be exactly $\bar{b}(i)$ blue points in each vertex $i$ and each of these will be an in-point or an out-point and the end of a path in $F_t$. The matching $M_F$ is given by the pairs of endpoints of the paths in $F_t$ when the algorithm stops, with any remaining free red points matched at random.

Note that a matched pair in $M_F$ is determined exactly when a path in $F_t$ from a red or blue point to another red or blue point is completed, that is, when a pair $f_1f_2$ in $B$ that completes a path is exposed. Hence the distribution of $M_F$ depends only on the distribution of the out-point $f_2$ in these instances, which is uniform on all free points. The choice of $f_1$ does not matter, nor does the choice of $f_2$ when $f_1f_2$ does not complete a path in $F_t$.

To achieve $M_F$ the colouring and linking is done as follows at each step of $\text{ALoad}(k)$.

First, $\text{ALoad}(k)$ exposes a point that will become an in-point. At some time, $t$, let $f_1$ be this point. Let $i_1$ be the vertex containing $f_1$ and let $\bar{b}_1$ denote $\bar{b}(i_1)$. Let $b_1$ be the number of blue points (both free and exposed) in $i_1$ at this time. Now, we use the fact that in any vertex with $u$ uncoloured points and $b$ blue points

$$u - (\bar{b} - b)$$

is even and non-negative. \hfill (A)

This is proved in the next lemma.

**Step 1**

$c_{1.1}$: If $f_1$ is uncoloured and $b_1 < \bar{b}_1$, colour $f_1$ blue.

$c_{1.2}$: If $f_1$ is uncoloured and $b_1 = \bar{b}_1$, colour $f_1$ black. If there are no white points in $i_1$ then link $f_1$ to an uncoloured point $q$ in $i_1$ and colour $q$ white. By property (A) such a point $q$ exists. Otherwise, choose any white point $q_w$ in $i_1$, which must be
linked to a black point $q_b$ in $i_1$. Swap the linking so that $f_1$ is linked to $q_b$ and $q_w$ is not linked. Change $q_w$ to uncoloured.

c_{1.3}:\text{ If } f_1 \text{ is white, colour } f_1 \text{ black.}

c_{1.4}:\text{ If } f_1 \text{ is red or blue do nothing.}

Next $A\text{Load}(k)$ exposes the mate $f_2$ of $f_1$, which becomes an out-point in step $t$. Let $i_2$ be the vertex containing $f_2$ and let $\bar{b}_2$ denote $\bar{b}(i_2)$. Let $b_2$ be the number of blue points (both free and exposed) in $i_2$ and let $u_2$ be the number of uncoloured points in $i_2$ at this time.

\textbf{Step 2}

c_{2.1}:\text{ If } f_2 \text{ is uncoloured, colour } f_2 \text{ blue with probability } \frac{\bar{b}_2-b_2}{u_2}. \text{ Otherwise colour } f_2 \text{ black and link } f_2 \text{ to any uncoloured point } q \text{ in } i_2. \text{ By property (A) such a point } q \text{ exists. Colour } q \text{ white.}

c_{2.2}:\text{ If } f_2 \text{ is white, colour } f_2 \text{ black.}

c_{2.3}:\text{ If } f_2 \text{ is red or blue do nothing.}

\textbf{Step 3} \text{ If the number of blue points in } i_2 \text{ plus the number of uncoloured points in } i_2 \text{ equals } \bar{b}_2 \text{ then colour all remaining uncoloured points in } i_2 \text{ blue.}

It is clear from this that within each vertex $v$, white and black points are always linked in pairs, and that the number of blue points in $v$ is never greater than $\bar{b}(v)$.

We now interrupt the proof of Lemma 7.1 to formally define $M_F$ and show some of its properties. (End of proof part 1.) \hfill \eop

\textbf{Definition 7.4} Define $M_F$, the random matching induced by $F_t$, as follows. Any two endpoints of a path in $F_t$ are matched in $M_F$ provided they are both either blue or red. Additionally, any red points which are not ends of paths at time $T_h$ are paired uniformly at random and matched in $M_F$. 

153
Note that any surviving red point at time $T_h$ must be matched to a red point in $M_F$. That $M_F$ contains $X_T$ blue points, all of which are in-points and out-points, and $H_T$ red points follows from the next lemma.

**Lemma 7.5** At time $T_h$, each vertex $i$ contains no white points and exactly $\bar{b}(i)$ blue points, all of which are in-points or out-points.

**Proof.** Consider vertex $i$. There is nothing to show if $i$ contains red points, nor if $i$ is a priority 1 vertex since in those vertices all $\bar{b}$ free points are coloured blue at time $T$ and will certainly be in-points or out-points at time $T_h$.

Assume that $i$ contains no red points and is not a priority 1 vertex. So

$$\bar{b}(i) = 2k - d_T(i) < d_T(i) - 2j_T(i).$$

Let $u$, $b$, $b_f$ and $w$ be the numbers of uncoloured points, blue points, free blue points and white points respectively in $i$ at some time $T \leq t \leq T_h$. Note that the free points in $i$ are the uncoloured points, the white points and the free blue points.

The proof follows from two statements:

$$u - (\bar{b} - b)$$

is even and non-negative, \hspace{1cm} (A)

$$2k - d^L(i) \geq \bar{b} - b + b_f + w.$$ \hspace{1cm} (B)

Recall also that $b \leq \bar{b}$.

First, (A): Let $J = u - (\bar{b} - b)$. At time $T$, $b = 0$ so

$$u - (\bar{b} - b) = d_T(i) - 2j_T(i) - \bar{b}$$

$$= 2k - \bar{b} - 2j_T(i) - \bar{b}$$

$$= 2k - 2\bar{b} - 2j_T(i),$$

which is even and positive since $d_T(i) - 2j_T(i) > \bar{b}.$
Assume $J$ is even and positive at some time $t_1 \geq T$. If, at this time, an uncoloured point becomes an in-point or an out-point and is then coloured blue according to $c_{1,1}$ or $c_{2,1}$, both $u$ and $\bar{b} - b$ decrease by one so $J$ remains the same.

If an uncoloured point becomes an in-point, according to $c_{1,2}$, it can only be coloured black if $\bar{b} = b$. In this case $u \geq 2$ before colouring since $J$ is positive and even. If $w = 0$ then $u$ decreases by two since one point becomes black and one becomes white. If $w > 0$ then $u$ does not change since one point becomes black and one white point becomes uncoloured. Hence $J$ is either even or zero after colouring.

If an uncoloured point becomes an out-point, according to $c_{2,1}$ it can only be coloured black if $\bar{b} - b < u$. In this case $u - (\bar{b} - b) \geq 2$ before colouring since $J$ is even. One point in $v$ becomes black and one becomes white so $u$ decreases by two and $J$ is still even or zero.

In $c_{1,3}$, $c_{1,4}$, $c_{2,2}$ and $c_{2,3}$ there is no change to $u$ or $b$ so $J$ remains the same.

Finally, if $J$ is zero at any time then $u = \bar{b} - b$ and according to step 3 all the uncoloured points will get coloured blue. After this both $u$ and $\bar{b} - b$ are zero so $J$ remains zero from this time on.

Hence by induction at all times $J$ is even and positive, or zero, and (A) holds.

Next, (B): At time $T$, $b$, $b_f$ and $w$ are zero and $\bar{b} = 2k - d_T(i)$ by definition so (B) holds. Assume (B) holds at time $t_1 \geq T$.

If an uncoloured point becomes an in-point or out-point and is coloured blue according to $c_{1,1}$ or $c_{2,1}$, then $b$ increases by one and $d$ either increases by one or decreases by one. In either case (B) still holds.

Similarly if a free blue point or a white point becomes an in-point or an out-point then, according to $c_{1,3}$, $c_{1,4}$, $c_{2,2}$ and $c_{2,3}$, $b_f + w$ will decrease by one (the white point will become a black point) whilst $d$ either increases by one or decreases by one. In all cases (B) still holds.

If $\bar{b} - b = 0$ and an uncoloured point becomes an in-point then according to $c_{1,2}$ it will
be coloured black. Note that \( b_f = 0 \) and \( 2k - d^L(i) > 0 \) in this case. If \( w > 0 \) then both sides of (B) will decrease by one since a white point will become uncoloured, and (B) will still hold. If \( w = 0 \) then \( d \) equals \( u \) plus twice the number of in-points. Now \( u \) is even since \( J \) is even, so \( d \) is even also. Hence \( 2k - d^L(i) \geq 2 \) and the right hand side of (B) is zero before this step. After an uncoloured point becomes a black in-point and a second uncoloured point becomes white \( 2k - d^L(i) \geq 1 \) and the right hand side of (B) is one so (B) still holds.

If an uncoloured point becomes an out-point and is coloured black according to \( c_{2,1} \), then both sides of (B) increase by one so it still holds.

If \( \tilde{b} - b = u \). then in Step 3, \( b_f \) and \( b \) both increase by \( u \) so the right hand side of (B) remains unchanged, as does the left hand side.

Hence by induction (B) holds for all \( T \leq t \leq T_h \).

At time \( T_h \) either \( 2k - d^L(i) = 0 \) or there are no free points remaining in \( i \).

Let \( t = T_h \) and suppose \( 2k - d^L(i) = 0 \). Then since (B) holds and all terms are non-negative, \( \tilde{b} - b = 0 \), \( b_f = 0 \) and \( w = 0 \).

Otherwise, suppose there are no free points in \( i \). Then \( u \), \( w \) and \( b_f \) are each zero. Since (A) is non-negative, \( \tilde{b} - b \) is zero also.

Thus in either case there are no white points remaining in \( i \) and exactly \( \tilde{b} \) blue points, each of which is an in-point or an out-point as required.

Before proceeding with the proof of Lemma 7.1, the probability that any red point is paired to any another red point in \( M_F \) is needed. It will become clear in the following that the matching \( M_F \) occurs u.a.r. subject to the point being matched. The argument will produce the Lemma 7.6 below, which gives the precise statement required for the proof of Lemma 7.1.

**Lemma 7.6** When \( M_F \) is fully determined, let \( RR \) be the set of red points which are matched to red points in \( M_F \). Then

\[ \text{Lemma 7.6} \]
(i) Let $Z$ be a random variable for the number of red points in red-red pairs in a random uniform matching of $X_T$ blue and $H_T$ red points. The size of $RR$ has the same distribution as $Z$.

(ii) Conditional upon the set $RR$, the matching of red points to red points in $RR$ occurs u.a.r.

(iii) $RR$ as a subset of all red points occurs u.a.r.

The proof will follow easily from the next lemma.

At any time $t$ after $T$, each blue or red point is either free, or is connected by a path in $F_t$ to a white point, or is connected by a path in $F_t$ to another red or blue point. Let $M_B(t)$ and $M_R(t)$ be the number of blue and red points respectively which are connected by a path in $F_t$ to another blue or red point (and so their matched pair in $M_F$ is already determined). Let $W_B(t)$ and $W_R(t)$ be the number of blue and red points respectively which are connected by a path in $F_t$ to a white point. Equivalently, these are the numbers of white points which are connected by a path in $F_t$ to a blue or red point respectively.

Let $r_f(i)$, $b_f(i)$ and $b(i)$ be the number of free red points, free blue points and blue points, respectively, in vertex $i$ at time $t$. Then

$$\sum_i b(i) = W_B(t) + M_B(t) + \sum_i b_f(i), \quad (7.2)$$

$$H_T = W_R(t) + M_R(t) + \sum_i r_f(i). \quad (7.3)$$

Say that a point $p$ represents a blue point if either $p$ is blue, or $p$ is white and there is a path in $F_t$ from $p$ to a blue point. Let $p \rightarrow$ blue mean “$p$ represents a blue point”. Similarly, $p \rightarrow$ red has the analogous meaning, “$p$ represents a red point”.

**Lemma 7.7** Suppose that $f_1$ is the point chosen at time $t$ to become an in-point and that after colouring and linking according to step 1, the mate, $f_2$, of $f_1$ in $\hat{A}(t)$ is coloured and linked according to step 2. Then in $F_t - (f_2,f_1)$, with $L$ the number of free points
remaining at this time and assuming that \( f_1 \) is counted in \( M_B \) or \( M_R \) if it represents a blue or red point respectively,

\[
P(f_2 \rightarrow \text{blue}) = \frac{X_T - M_B}{L}, \quad (7.4)
\]
\[
P(f_2 \rightarrow \text{red}) = \frac{H_T - M_R}{L}. \quad (7.5)
\]

Moreover, conditioning on the event \([f_2 \rightarrow \text{red}]\), the probability that \( f_2 \) represents a particular red point \( r \) is uniform:

\[
P(f_2 \text{ represents } r) = \frac{1}{H_T - M_R}. \quad (7.6)
\]

**Proof.** For each unmatched red point, \( r \), in \( \hat{A}(t) \), there is exactly one representative, which is either \( r \) itself, if \( r \) is free, or is a white point connected to \( r \) by a path in \( F_t \). Hence, the total number of free points representing unmatched red points is

\[
W_R + \sum_i r_f(i) = H_T - M_R,
\]

by (7.3).

Since \( f_2 \) is uniformly distributed on all free points, (7.5) and (7.6) follow.

Similarly, the total number of free points representing unmatched blue points is

\[
W_B + \sum_i b_f(i),
\]

so the probability that \( f_2 \) represents a blue point before step 2 is

\[
P(f_2 \rightarrow \text{blue before step 2}) = \frac{W_B + \sum_i b_f(i)}{L}.
\]

Additionally, \( f_2 \) may represent a blue point after step 2 if it is uncoloured and gets coloured blue according to \( c_{2.1} \). The probability of this is

\[
P(f_2 \text{ gets coloured blue}) = \frac{1}{L} \sum_{i: u(i) > 0} \frac{t(i) - b(i)}{u(i)}
\]
\[
= \frac{1}{L} \sum_i \frac{b(i) - b(i)}{u(i)}
\]
\[
= \frac{X_T - W_B - M_B - \sum_i b_f(i)}{L},
\]

158
using (7.1) and (7.2). Hence
\[
\Pr(f_2 \to \text{blue}) = \frac{W_B + \sum_i b_f}{L} + \frac{X_T - W_B - M_B - \sum_i b_f}{L} = \frac{X_T - M_B}{L},
\]
as required. \[\blacksquare\]

**Proof of Lemma 7.6.** Part (i) follows from (7.4) and (7.5), since conditioning on \(f_2\) representing a red or blue point, these become the same probabilities as if \(X_T\) blue and \(H_T\) red points were matched uniformly at random. Parts (ii) and (iii) follow from (7.6) since any unmatched red point is equally likely to be represented by \(f_2\) in any step of \(\text{ALoad}(k)\). Further details are omitted. \[\blacksquare\]

Next we define a partial pairing and a graph from \(\hat{A}(T_h)\) and \(M_F\). In the ensuing lemmas we establish some properties of this graph, from which the proof of Lemma 7.1 will follow.

Let \(P_R\) be the partial pairing consisting of all the red points in \(\hat{A}(T_h)\), and with pairs in \(P_R\) being those matched in \(M_F\) (i.e. those in \(RR\)). Let \(G_R\) be the pseudograph obtained by identifying all points of \(P_R\) in the same vertex of \(\hat{A}(T_h)\). Pairs in \(P_R\) become edges in \(G_R\).

Now we examine which points in \(P_R\) have survived \(\text{ALoad}(k)\). First of all, the points not in \(RR\) have not survived, since they are matched with blue points in \(M_F\). Secondly, some of the red-red pairs in \(M_F\) may have arisen from paths in \(F_t\) and if so, have not survived either.

It will be shown that deterministically, only red points in the 2-core of \(G_R\) can have survived. Then it will be shown that, if \(X_T\) is much larger than \(Z_T\), \(G_R\) is acyclic with high probability, from which it follows that the 2-core is almost certainly empty.

**Lemma 7.8** All surviving red points at at time \(T_h\) are in the 2-core of \(G_R\).

**Proof.** All surviving red points in \(\hat{A}(T_h)\) are in vertices of load-degree \(2k\) or greater. Let \(i\) be a vertex containing surviving red points. If \(d^L(i) > 2k\) then there are no in-points in

159
and at least 2\(k+1\) surviving red points. If \(d^L(i) = 2k\) then the number of surviving red points in \(i\) is \(2k - 2d^L(i)\) which is even or zero.

Hence, it is not possible for \(i\) to contain a single surviving red point.

With respect to \(G_R\), this means that any point in a vertex of degree one in \(G_R\) has not survived. If a point \(p\) has not survived, then the pair of \(p\) has not survived either, so if \(v\) is a vertex of degree \(d\) in \(G_R\) and at least \(d - 1\) neighbours of \(v\) have not survived, then no point in \(v\) has survived. Hence, to remove vertices in \(G_R\) in which no point has survived, vertices of degree one in \(G_R\) may be successively deleted until no more remain. This will leave only the 2-core of \(G_R\), so any surviving red points must be in the 2-core.

**Definition 7.9** For an allocation, \(A\), Let \(Z\) be the degree of a randomly chosen vertex of degree no less than \(j\) in \(A\), and, for a randomly chosen point \(p\) in a vertex of degree no less than \(j\) in \(A\), let \(Y\) be the number of points in the same vertex as \(p\).

Define \(\mathcal{U}_j(A)\) to be the event that there exists constants \(\tilde{\lambda}, c_1\) and \(c_2\) and truncated Poisson distributed random variables \(\tilde{Z}\) and \(\tilde{Y}\) with

\[
P(\tilde{Z} = d) = \begin{cases} 
  c_1 \tilde{\lambda}^d / d! & \text{for } d \geq j \\
  0 & \text{otherwise},
\end{cases}
\]

\[
P(\tilde{Y} = d) = \begin{cases} 
  c_2 d \tilde{\lambda}^d / d! & \text{for } d \geq j \\
  0 & \text{otherwise},
\end{cases}
\]

such that \(Z\) and \(Y\) are stochastically dominated by \(\tilde{Z}\) and \(\tilde{Y}\):

\[
P(Z \geq r) \leq P(\tilde{Z} \geq r), \quad \forall r \geq 0,
\]

\[
P(Y \geq r) \leq P(\tilde{Y} \geq r), \quad \forall r \geq 0.
\]  

(7.7)

Note that writing \(\tilde{Z}\) as \(\tilde{Z}_j\), \(\tilde{Y}\) has the same distribution as \(\tilde{Z}_{j-1}\). Both \(\tilde{Z}\) and \(\tilde{Y}\) have an exponential tail.

It’s clear that \(c_1 = 1/f_j(\tilde{\lambda})\) and \(E(\tilde{Z}_j) = \frac{\hat{\lambda} f_{j-1}(\hat{\lambda})}{f_j(\hat{\lambda})} = \psi_j(\hat{\lambda})\). Note also that \(E(\tilde{Z}_{j-1}) < \)
\[ E(\tilde{Z}_j) \quad \text{and} \quad \begin{align*}
E(\tilde{Z}_j(\tilde{Z}_j - 1)) &= \sum_{d \geq j} \frac{d(d-1)\lambda^d}{f_j(\lambda)d!} \\
&= \frac{\lambda^2f_{j-2}(\lambda)}{f_j(\lambda)} \\
&= E(\tilde{Z}_j)E(\tilde{Z}_{j-1}) \\
&< E(\tilde{Z}_j)^2. \tag{7.8}
\]

**Lemma 7.10** Assume \( W_T, H_T > \epsilon n \) for some \( \epsilon > 0 \). Then a.a.s. \( U_{2k+1}(\hat{A}_T) \).

**Proof.** Let \( Z \) and \( Y \) be as in Definition 7.9 (with \( j = 2k + 1 \)). We will show that there exists \( \hat{\lambda} \) such that with \( c_1 = (f_{2k+1}(\hat{\lambda}))^{-1} \) and \( c_2 = (\hat{\lambda}f_{2k}(\hat{\lambda}))^{-1} \), and \( \tilde{Z}, \tilde{Y} \) as in Definition 7.9, \( Z \) and \( Y \) are stochastically dominated by \( \tilde{Z}, \tilde{Y} \) respectively.

First we note the following. For \( r \geq 2k + 2 \), let \( \varphi_r(\mu) := \frac{f_r(\mu)}{f_{2k+1}(\mu)} \). Then

\[
\frac{d\varphi_r(\mu)}{d\mu} = \frac{f_{r-1}(\mu)}{f_{2k+1}(\mu)} \left( 1 - \frac{f_r(\mu)f_{2k}(\mu)}{f_{r-1}(\mu)f_{2k+1}(\mu)} \right) \\
= \frac{f_{r-1}(\mu)}{f_{2k+1}(\mu)} \left( 1 - \frac{\psi_{2k+1}(\mu)}{\psi_r(\mu)} \right).
\]

For fixed \( \mu \), \( \psi_l(\mu) \) increases with \( l \) by Lemma 3.5 part (vi), so this is strictly positive and hence \( \varphi_r(\mu) \) increases with \( \mu \). Further, \( f_{r-1}(\mu) > f_r(\mu) \) so, with \( c(\mu) := 1 - \psi_{2k+1}(\mu)/\psi_{2k+2}(\mu) \),

\[
\frac{d\varphi_r(\mu)}{d\mu} > c(\mu)\varphi_r(\mu).
\]

For fixed \( \epsilon > 0 \), let \( \bar{\mu} = \mu + \epsilon \) and let \( c = \min_{\mu \leq \lambda \leq \bar{\mu}} c(\lambda) \). Then \( c > 0 \) and in the range \( \mu \leq \lambda \leq \bar{\mu} \)

\[
\frac{d\varphi_r(\mu)}{d\mu} > c\varphi_r(\mu).
\]

As already observed, \( \varphi_r(\mu) \) increases with \( \mu \), so this lower bound also increases with \( \mu \), and

\[
\varphi_r(\bar{\mu}) > \varphi_r(\mu) + c\varphi_r(\mu)\epsilon \\
= \varphi_r(\mu)(1 + c\epsilon). \tag{7.9}
\]
The high degree sequence of \( \hat{A}_T \) has the distribution of \( \text{Multi}(2k + 1, W_T, H_T) \), so by Corollary 3.10, for \( d \geq 2k + 1 \)

\[
P(Z = d) = \frac{\lambda^d}{f_{2k+1}(\lambda)d!}(1 + o(1)),
\]

where \( \lambda = \lambda_{2k+1,W_T/H_T} \), the positive root of \( \psi_{2k+1}(\lambda) = H_T/W_T \). Hence, for \( r \geq 2k + 2 \)

\[
P(Z \geq r) = \sum_{d \geq r} \frac{\lambda^d}{f_{2k+1}(\lambda)d!}(1 + o(1))
= \frac{f_r(\lambda)}{f_{2k+1}(\lambda)}(1 + o(1))
= \phi_r(\lambda)(1 + o(1)).
\]

For \( \epsilon > 0 \) let \( \tilde{\lambda} = \lambda + \epsilon \), and let \( \tilde{Z} \) be the truncated Poisson distribution random variable with parameter \( \tilde{\lambda} \). Then there exists constant \( c > 0 \), independent of \( r \), such that

\[
P(\tilde{Z} \geq r) = \sum_{d \geq r} \frac{\tilde{\lambda}^d}{f_{2k+1}(\lambda)d!}(1 + o(1))
= \frac{f_r(\tilde{\lambda})}{f_{2k+1}(\lambda)}(1 + o(1))
= \phi_r(\tilde{\lambda})(1 + o(1)).
\]

by (7.9). Hence, a.a.s. \( P(\tilde{Z} \geq r) \geq P(Z \geq r) \) for all \( r \geq 2k + 2 \). The cases \( r \leq 2k + 1 \) are trivial as \( P(\tilde{Z} \geq r) = P(Z \geq r) = 1 \).

A similar argument shows that \( Y \) is stochastically dominated by \( \tilde{Y} \).

**Lemma 7.11** There exists a function \( g \) with \( g(x) \to 0 \) as \( x \to \infty \) such that if \( X_T, H_T > \epsilon n \) for some \( \epsilon > 0 \) and \( U_{2k+1}(\hat{A}_T) \) holds, then for \( n \) sufficiently large

\[
P(G_R \text{ has a cycle}) < g \left( \frac{X_T}{H_T} \right).
\]

**Proof.** Fix \( \hat{A}_T \), in particular, this means conditioning on \( X_T, H_T \) and the degree sequence of the vertices of degree greater than \( 2k \) at time \( T \). Assume that \( X_T, H_T > \epsilon n \) for some
\( \epsilon > 0 \) and \( \mathcal{U}_{2k+1}(\hat{A}_T) \) holds. Let \( \kappa = X_T/H_T \). The following applies for \( \kappa \) sufficiently large. Let \( Y, Z, \tilde{Y} \) and \( \tilde{Z} \) be as in Definition 7.9, with \( j = 2k + 1 \).

The partial pairing \( P_R \) may be generated by starting with \( X_T \) artificial blue points and the \( H_T \) red points grouped in their vertices of \( A_T \). By Lemma 7.6 and further to the discussion of Section 7.1, \( P_R \) can be generated with the correct distribution by taking at each step a red point \( r \) and pairing it with another point \( x \) chosen u.a.r. from the remaining red and blue points. Call this process the \( RB \) process, and the pairing it generates \( P_{RB} \). If blue points are discarded from \( P_{RB} \) and red points with blue mates in \( P_{RB} \) are considered unpaired, the resulting partial pairing has the same distribution as \( P_R \).

Let \( m = H_T + X_T \) and let \( q = \frac{H_T}{m} \). Then \( m > 2\epsilon n \) and \( q = \frac{1}{\kappa + 1} \). For each red point \( y \) let \( d_y \) be the number of points in the vertex containing \( y \), before the start of the \( RB \) process. Define \( \mu \) to be

\[
\mu := \mathbb{E}(Y - 1) = \sum_y \frac{d_y - 1}{H_T}.
\]

\( \mathbb{E}(\tilde{Y}) \) is bounded so it follows from \( \mathcal{U}_{2k+1}(\hat{A}_T) \) that \( \mathbb{E}(Y) \) is bounded also. Hence \( \mu \) is bounded. Note that, by comparison with (7.8), and since \( \tilde{Z} \) stochastically dominates \( Z \) and \( \mathbb{E}(Z) = H_T/W_T \)

\[
\sum_y (d_y - 1) = \sum_i d(i)(d(i) - 1)
\]

\[
= W_T \mathbb{E}(Z(Z - 1)) < W_T \left( \left( \frac{H_T}{W_T} \right)^2 + o(1) \right) = O(kH_T). \tag{7.10}
\]

For the first red point paired in the \( RB \) process, the probability that its mate \( x \) is red is

\[
P(x \text{ is red}) = \frac{H_T - 1}{m - 1} = q + O\left( \frac{1}{m} \right). \tag{7.11}
\]

If \( x \) is red, then the expected number of other points in the same vertex is

\[
\mathbb{E}(d_x - 1) = \sum_y \frac{d_y - 1}{H_T - 1} = \mu + O\left( \frac{1}{m} \right). \tag{7.12}
\]

163
Consider the following branching process derived from the RB process. Start with a randomly chosen red point $r$ and expose the pair $rx$. Let $X_0 = 1$ and for each step $t \geq 1$ (in the RB process, not in $\text{ALoad}(k)$), the number of children in the $t$th step is defined to be

$$X_t = \begin{cases} 
0 & \text{if } x \text{ is blue} \\
 d_x(t) - 1 & \text{if } x \text{ is red},
\end{cases}$$

where for each red point $y$, $d_y(t)$ is the number of unmatched points in the same vertex as $y$ before the $t$th step of the RB process. Certain variables in the RB process will relate to those in this simple process. In particular, a bound on the size of this branching process will be used to obtain a bound on the probability that $G_R$ contains a cycle.

Let $Y_t = d_y(t)$ for a randomly chosen red point $y$. Then, as in (7.12),

$$E(Y_t - 1) \leq \sum_x \frac{d_x(t) - 1}{H_T - 2t}.$$ 

Now, using (7.11) and the definition of $X_t$,

$$E(X_1) = E(Y_1 - 1) \frac{(H_T - 1)}{m - 1}.$$ 

So, for some $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, we have, for $t < \log^2 m$,

$$E(Y_t - 1) < \mu(1 + \varepsilon_2) \quad \text{and} \quad E(X_t) < p, \quad (7.13)$$

with

$$p := q(1 + \varepsilon_1)\mu(1 + \varepsilon_2).$$

Recall the definition of $\tilde{Y}$ and that $Y$ is stochastically dominated by $\tilde{Y}$. Note from the proof of Lemma 7.10 that if $\tilde{\lambda}$ is increased slightly, then $\tilde{Y}$ will still stochastically dominate $Y$ (we show this for $Z$, and it’s easily seen to be true for $Y$ also). For $t < \log^2 m$, $Y_t$ stays close to $Y$, hence it’s possible to choose $\tilde{\lambda}$ so that each $Y_t$ is stochastically dominated by $\tilde{Y}$. Further, it’s possible to choose $\varepsilon_2$ and $\tilde{\lambda}$ so that, in addition, $E(\tilde{Y} - 1) = \mu(1 + \varepsilon_2)$.

Hence, we may assume that for $t < \log^2 m$, $Y_t$ is stochastically dominated by $\tilde{Y}$:

$$P(Y_t \geq r) \leq P(\tilde{Y} \geq r) \quad \forall r \geq 0,$$
and \( \mathbf{E}(\tilde{Y} - 1) = \mu(1 + \varepsilon_2) \). Note that

\[
p = \frac{\sum_y (d_y - 1)}{m} (1 + \varepsilon_1)(1 + \varepsilon_2) = O\left(\frac{kH_T}{m}\right)
\]  

(7.14)

by (7.10).

Let \( L \) be the lifetime of the branching process, and let \( W \) be the size of a Galton-Watson process with children distributed as \( \tilde{X} \) where

\[
\tilde{X} = \begin{cases} 
\tilde{Y} - 1 & \text{with probability } q(1 + \varepsilon_1) \\
0 & \text{otherwise.}
\end{cases}
\]

Then \( \mathbf{E}(\tilde{X}) = p \) and

\[
\mathbf{P}(X_t \geq r) \leq \mathbf{P}(\tilde{X} \geq r) \quad \forall r \geq 0,
\]

\[
\mathbf{P}(L \geq s) \leq \mathbf{P}(W \geq s) \quad \forall s \geq 0
\]

and

\[
\mathbf{E}(W) = p \left(\frac{1}{1 - p}\right).
\]  

(7.15)

\( \tilde{Y} \) has an exponential tail, so \( \tilde{X} \) does also, and by Theorem 1.4, \( W \) has one also. That is, there exist \( k_1 \) and \( k_2 \) such that \( \mathbf{P}(W \geq s) < k_1 \exp(-k_2s) \). So for \( c \) and \( m \) sufficiently large

\[
\mathbf{P}(L \geq c \log m) \leq \mathbf{P}(W \geq c \log m)
\]

\[
= o\left(\frac{1}{m}\right), \quad \text{assuming } c > \frac{1}{k_2}.
\]  

(7.16)

Let \( i_0 \) be a vertex containing \( d \) free points and condition on \( i_0 \) containing the first red point exposed in the \( RB \) process. The aim in what follows is to bound the probability that \( i_0 \) is in a cycle.

Let \( H_t \) be the event that in the \( t \)th step the mate of the red point exposed in that step is, for the first time, in \( i_0 \). So \( H_t \) is the event that a cycle including \( i_0 \) first appears in the
$t$th step. Let $C_t$ be the number of unexposed children in the branching process before the $t$th step and let $I_t$ be an indicator variable for the event $[C_t > 0]$. Then

$$\Pr(H_t) \leq \frac{(d-1)\Pr(I_t = 1)}{m - 2t - 1} \quad \text{and}$$

$$\sum_{t \geq 1} \Pr(I_t = 1) = \mathbb{E}(L) \leq \frac{p}{1 - p} \quad \text{by (7.15)}.$$

Summing over $t$ to get the probability of a cycle including $i_0$, for $m$ sufficiently large

$$\Pr(i_0 \text{ is in a cycle}) = \sum_{t \geq 1} \Pr(H_t)$$

$$< \sum_{t = 1}^{e \log m} \Pr(H_t) + \Pr(L \geq c \log m)$$

$$< \sum_{t = 1}^{e \log m} \frac{d \Pr(I_t = 1)}{m - 2t - 1} + o\left(\frac{1}{m}\right) \quad \text{by (7.16)}$$

$$< \frac{2d}{m} \mathbb{E}(L) + o\left(\frac{1}{m}\right)$$

$$< \frac{2dp}{m(1 - p)} + o\left(\frac{1}{m}\right)$$

$$< \frac{4dp}{m} + o\left(\frac{1}{m}\right).$$

The first point $r$ in the branching process is chosen at random so for any vertex $i$ of degree $d(i)$, conditioning on $r$ being in $i$, the probability that $i$ is in a cycle satisfies

$$\Pr(i \text{ is in a cycle}) < \frac{4d(i)p}{m} + o\left(\frac{1}{m}\right).$$

So, for $m$ sufficiently large

$$\mathbb{E}(\text{number of cycles in } G_R) < \mathbb{E}(\text{number of vertices in cycles in } G_R)$$

$$= \sum_{i \in G_R} \Pr(i \text{ is in a cycle})$$

$$< o(1) + \sum_{i \in G_R} \frac{4d(i)p}{m}$$

$$= o(1) + \frac{4H_T p}{m}$$

166
by (7.14). Hence, by Markov’s inequality

\[ P(G_R \text{ has a cycle}) < O\left(\frac{k}{\kappa^2}\right). \]

(7.17)

Recall that this holds for all large \( \kappa \), and the lemma follows. Recall that this holds for any \( \kappa \) sufficiently large.  

Now we return to the proof of Lemma 7.1.

**Proof of Lemma 7.1, part 2.** Consider the matching \( M_F \) and the pseudograph \( G_R \) obtained from \( M_F \) and \( \hat{A}(T_h) \). If \( G_R \) is acyclic then it has no 2-core so by Lemma 7.8, \( H(T_h) = 0 \) and \( Z(T_h) = 0 \). Hence

\[ P(Z(T_h) = 0) > 1 - P(G_R \text{ has a cycle}). \]

Recall that \( H_T = Z_T + 2kW_T \) and \( Z_T \geq W_T \), so

\[ H_T \geq (2k + 1)Z_T, \]

and

\[ X_T > \xi Z_T \]

\[ \geq \frac{\xi}{2k + 1} H_T. \]

For an a.a.s. result we may assume that \( U_{2k+1}(\hat{A}_T) \) holds, since by Lemma 7.10 it holds a.a.s. Hence, by Lemma 7.11 and (7.17)

\[ P(G_R \text{ has a cycle}) < O\left(\frac{k}{(\xi/(2k + 1))^2}\right). \]

Recall that \( \xi = K/\delta^{\frac{1}{2}} \), so for \( K \) and \( n \) sufficiently large the right hand side is less than \( \delta \), and

\[ P(Z(T_h) = 0) > 1 - \delta. \]

This completes the proof of Lemma 7.1.  

167
7.3 Proof of Lemma 6.3 and Theorem 6.1

**Proof of Lemma 6.3.** For fixed $\epsilon > 0$ and integer $k \geq 2$, let $M \in \mathcal{M}(n_0, m_0)$ and let $M_c$ be the $(k + 1)$-core of $M$. Let $\rho$ be the density of $M_c$, and let $\hat{B}$ be the partial pre-allocation corresponding to $M_c$. Assume that $\rho < 2k - \epsilon$.

Let $X(t)$ and $Z(t)$ be random variables for the process $\Omega_{\text{ALoad}(k)}$ determined by $\text{ALoad}(k)$ applied to $\hat{B}(0) = \hat{B}$, as defined by (6.3) and (6.4). It follows from Lemma 6.6 that $X(0) > Z(0) + \epsilon n$.

Lemmas 6.15, 6.17 and 6.18 together imply (6.6): For any constants $\xi > 0$ and $\epsilon > 0$, there exists $\omega > 0$ and $\epsilon_c > 0$ such that a.a.s. if $X(0) > Z(0) + c n$ then $X > \xi Z + \epsilon_c n$ before time $T_f - \omega n$.

Lemma 7.1 states that for any $\delta > 0$ there exists $\xi = \xi(\delta)$ such that for all $\epsilon_c, \omega > 0$, a.a.s. if there exists $T < T_f - \omega n$ such that $X(T) > \xi Z(T) + \epsilon_c n$ then $P(Z(T_h) = 0) > 1 - \delta$.

The combination of these two results implies that for any $\delta, \epsilon > 0$, a.a.s. if $X(0) > Z(0) + \epsilon n$ then $P(Z(T_h) = 0) > 1 - \delta$. Hence $P(Z(T_h) = 0)$ is lower bounded by values arbitrarily close to 1. So a.a.s. $Z(T_h) = 0$ and by Lemma 6.9, $\text{ALoad}(k)$ has a successful output a.a.s.  

**Proof of Theorem 6.1.**

Recall that

$$h_{k+1}(\mu) := \frac{\mu e^{\mu}}{f_k(\mu)},$$

$$\rho_k := h_{k+1}(\lambda_k),$$

$\lambda_k$ is defined by

$$\psi_{k+1}(\lambda_k) = 2k,$$

and $\mu_{k+1,\rho}$ is the root of

$$h_{k+1}(\mu_{k+1,\rho}) = \rho.$$

Assume that $\rho < \rho_k - \epsilon_1$ for some $\epsilon_1 > 0$. Then there exist positive $\epsilon_2, \epsilon_3 = O(\epsilon_1)$ such
that the following are equivalent:

\[ \rho < \rho_k - \epsilon_1, \]
\[ h_{k+1}(\mu_{k+1,\rho}) < h_{k+1}(\lambda_k) - \epsilon_1, \]
\[ \mu_{k+1,\rho} < \lambda_k - \epsilon_2, \]
\[ \psi_{k+1}(\mu_{k+1,\rho}) < \psi_{k+1}(\lambda_k) - \epsilon_3 = 2k - \epsilon_3, \]

Note that this relies on the derivatives of \( \psi_{k+1}(\mu) \) and \( h_{k+1}(\mu) \) being positive for \( \mu \geq \mu_{k+1,\rho} \). By Lemma 3.5 the derivative of \( \psi_{k+1}(\mu) \) is positive for all \( \mu > 0 \), and by Lemma 5.2 the derivative of \( h_{k+1}(\mu) \) is positive for all \( \mu \geq \mu_{k+1,\rho} \), since \( \rho > c_{k+1} \), the unique minimum of \( h_{k+1}(\mu) \).

By (5.20), the density of the \((k+1)\)-core of \( M \in \mathcal{M}(n, \frac{m}{n} \rho) \) is a.a.s. \( \psi_{k+1}(\mu_{k+1,\rho})(1+o(1)) \), and hence a.a.s. less than \( 2k - \epsilon \) for some \( \epsilon > 0 \).

Thus if \( \hat{B}_0 \) is the partial pre-allocation associated with the \((k+1)\)-core of \( M \), then by Lemma 6.3, \textbf{ALoad}(k) applied to \( \hat{B}_0 \) will have a successful output a.a.s. By Lemma 4.25, this implies that \textbf{Load}(k) applied to \( M \) will have a successful output a.a.s., and the same result for \( G \in \mathcal{G}(n, \frac{m}{n} \rho) \) follows from Lemma 3.11.

On the other hand, by Lemma 1.1, \textbf{Load}(k) cannot have a successful output if applied to a graph, or indeed a pseudograph, with a \((k+1)\)-core of density greater than \( 2k \). Assume that \( \rho > \rho_k + \epsilon_1 \) for some \( \epsilon_1 > 0 \). Then, using the same arguments as above with the inequalities reversed, there exists \( \epsilon_3 > 0 \) such that the following are equivalent:

\[ \rho > \rho_k + \epsilon_1, \]
\[ \psi_{k+1}(\mu_{k+1,\rho}) > \psi_{k+1}(\lambda_k) + \epsilon_3 \]
\[ \psi_{k+1}(\mu_{k+1,\rho}) > 2k - \epsilon_3. \]

So, by (5.20), the \((k+1)\)-core of \( M \), or \( G \), has density greater than \( 2k \) a.a.s. \( \blacksquare \)
Intuitively, we see that the optimality of Sanders’ algorithm is due, in part at least, to the choice of prioritisation. Reducing a graph to its \((k+1)\)-core is the best any such load balancing algorithm can do, in some sense: It is not possible to allocate any more edges than those not in the \((k+1)\)-core to the vertices not in the \((k+1)\)-core. By prioritising vertices for which the difference between load-degree and in-degree is \(k\) or less (priority 1 vertices), the algorithm ensures that as many edges as possible get allocated to these vertices, and there is no wastage. It’s less clear that the choice of prioritisation after this is crucial to the optimality. Certainly, it’s crucial to parts of the current proof, and it seems to keep the expected number of priority 1 vertices low, by next prioritising vertices which are closest to being priority 1 vertices. However, this is by no means conclusive.

An interesting problem to consider is whether choosing a different prioritisation scheme effects the optimality of Sanders’ algorithm. In particular, consider the obvious simplification in which, after priority 1 vertices, all vertices have the same priority. Is this also asymptotically almost surely optimal?

Another problem is to analyse the scaling window for optimality of Sanders’ algorithm, and how this is effected by the choice of prioritisation scheme.

Other examples exist of how incorporating a prioritisation step can convert a non-optimal
algorithm to an a.a.s. optimal one. Karp and Sipser [25], and later Aronson, Frieze and Pittel [1] investigated the optimality of an algorithm that finds a matching in a graph. It is known that a simple greedy algorithm that chooses edges uniformly at random to build the matching is not asymptotically optimal (does not find a maximal matching) on $G \in \mathcal{G}(n, \lceil \frac{cn^2}{2} \rceil)$. In [25] and with sharper estimates in [1], it is shown that a simple modification of this greedy algorithm results in a remarkable improvement: if edges adjacent to degree 1 vertices are prioritised and added to the matching first, this modified greedy algorithm is asymptotically optimal for $c < e$, or near optimal for $c > e$. Another example is in [36], where Shi and Wormald show that a random 4-regular graph a.a.s. has chromatic number 3. They do this by analysing a greedy algorithm that processes vertices according to a priority scheme, and showing that, with a suitable choice of scheme, it a.a.s. colours a random 4-regular graph with three colours. The right choice of priority scheme is one which prioritises certain vertices of ‘dangerous type’ to ensure that the number of these vertices stays low and the probability that the algorithm gets stuck with vertices it cannot colour, is minimised.

The success of these kinds of greedy algorithms, including Sanders’ algorithm, seems to rely on the identification and prioritising of certain vertices (or other parts of the graph) which will cause difficulties or wastage if not processed efficiently. One consequence of prioritisation is that the algorithms have the common property of going through phases, as discussed in Section 6.4. Typically, each phase is defined by the processing of vertices of a certain priority, and higher priority vertices have near zero expectation. A combination of differential equations to model the behaviour of some variables, and branching process arguments to bound the expectation of variables with near zero expectation, as used here and in [36], can provide an analysis of these algorithms. One possible research direction would be to seek further, general results on the application of these methods. Another direction could be to explore the relationship between optimality and the prioritisation of certain ‘critical’ vertices in algorithms for different kinds of graph problems.
Chapter 9

Greedy heuristics for Euclidean Steiner trees

In this final chapter we present some material on the development and analysis of heuristic algorithms for the $d$-dimensional Euclidean Steiner tree (EST) problem. The EST problem is, for a given set of terminals, to find the network spanning those terminals with minimum Euclidean length. The difference between this minimum network and the minimum spanning tree (MST), is that additional terminals may be added to the set to produce a shorter network.

The solution to the EST for a given set of terminals, $P$, is obviously a tree and is called the minimal Steiner tree (MST) on $P$. Some basic features of MSTs are outlined in Section 9.1. The original EST problem is in the plane, but can be generalised to higher dimensions where there are various applications such as mine tunnel optimisation and phylogenetic trees. The heuristics presented here can be used in any dimension.

The EST problem is known to be NP-Hard in all dimensions $d \geq 2$ [16], so low degree polynomial time heuristics are of particular interest. An obvious and fast heuristic is the minimum spanning tree. Heuristics which improve on the MST by adding extra vertices, called Steiner points, have been suggested by several authors (see [19]) and in 1992, Smith...
and Shor [37] introduced greedy heuristics based on MST algorithms, which we describe in Section 9.2. One aspect of the Steiner tree problem is in deciding exactly where to place the Steiner points for best effect. We improve on Smith and Shor’s greedy heuristics by allowing ourselves to reposition the Steiner points after and during the construction of a spanning network. In Section 9.3 a method for repositioning Steiner points is described. A framework for creating new heuristics along these lines is described in Section 9.4, and in Theorem 9.4 it is shown that, subject to a specific condition, such heuristics have guaranteed performance at least as good as the MST.

There is very little actual MSST data available for problems in Euclidean 3-space, except for small numbers of terminals ($N < 10$) where Smith’s exact algorithm [38] can be used. For larger $N$, the amount of memory and processing time required for this algorithm to run successfully become prohibitive.

We call a tree spanning a set of terminals and Steiner points a Steiner tree (StT). A typical approach to analysing heuristic performance is to compare StT length with MST length. However, the ratio of MST to MSST length is apparently highly variable so does not give a strong indication of how close the heuristic output is to the MSST. To counter this, we also compare the output of our heuristics to that of a metropolis algorithm (MA) which is described briefly in Section 9.4.4. Repeated trials of MA on the same terminal sets often result in the same StT, suggesting that the MSST may have been found in these cases. On the whole, MA almost always produced the shortest StT of all the heuristics tried, and did so in reasonable time, so presented a good point of comparison to indicate how close our heuristic outputs are to optimal.

We test the performance of our new heuristics by comparing their output to that of MA and the MST on sets of terminals distributed uniformly at random on the unit cube. It is worth noting that for real world applications, typical terminal sets may display certain phenomena that are not well represented by randomly generated terminal sets and visa-versa. For example, a real terminal set for a certain application may have fairly evenly spaced terminals, but terminals uniformly distributed on the unit $d$-cube will typically
appear to be clumped together if there is a sufficient number of them. Or, for a given application there may be certain constraints on the structure of the Steiner tree, and these constraints may greatly change the nature of the problem.

However, without a specific application in mind, testing our heuristics on random sets of terminals, and comparing their performance to that of the MST and the output of MA provides a good initial indication of how well they perform. Although there is little real data on MSTT lengths in Euclidean $d$-space for $d \geq 3$, in [3], the authors argue that their results on the length of a travelling-salesman tour extend to the MSTT: In particular, that the expected length of the MSTT for sets of terminals uniformly distributed in a bounded region is $\beta \sqrt{n}$, with the constant of proportionality, $\beta$, dependent only on $d$ and the size of the region. Moreover, the length of the MSTT is concentrated around its expectation as $n \to \infty$. Further investigations along these lines have been made by several authors, for the MST and variants of the Steiner tree problem as well as the travelling-salesman problem, although very little is know about the values of $\beta$ in most instances. One exception is in [6] where it is shown that in the plane, the value of $\beta$ for the rectilinear MSTT (the network with minimum length with respect to the rectilinear metric) is indeed less than the value of $\beta$ for the rectilinear MST, and it is conjectured that the same is true with respect to the Euclidean metric.

In Section 9.5 some results for our new heuristics for $d = 3$ are compared to both the MST and to the output of MA. On randomly generated sets of 10 and 50 terminals the best of the heuristics was found to find a StT less than 0.2% longer, on average, and less than 1% longer, 90% of the time, than that found using MA. The MST is, on average more than 5% longer than the output of MA. On the whole, our new heuristics present a good trade off between performance and running time if a fast polynomial time heuristic is sought.
9.1 Steiner tree basics

A tree spanning a set of terminals, $P$, and Steiner points, though not necessarily the shortest tree possible, shall be called a Steiner tree on $P$ ($StT(P)$). By the topology of a Steiner tree we are referring to the combinatorial structure, or the adjacencies of the terminals and Steiner points in the tree.

A very useful basic fact about MStTs is that their steiner points have degree 3 and the three adjacent edges meet at $120^\circ$ angles [17]. This is a result of the observations firstly, that replacing a vertex of degree greater than 3 by two or more degree 3 vertices can always decrease the length of the network, and secondly, that the sum of the lengths of edges incident on a vertex is always smallest when the angles between the edges are equal.

Now we shall restrict our definition of a Steiner tree to one where the steiner points have degree exactly three, giving us the following well known combinatorial properties: For $N$ terminals, there are up to $N - 2$ Steiner points in the StT and up to $N + (N - 2) - 1 = 2N - 3$ edges. A full topology is one with exactly this many edges and Steiner points. In a full topology all the terminals are leaves in the tree. Note that if we allow zero length edges, any topology can be replaced by a full topology. For reasons which will later be apparent we will assume that Steiner tree topologies in the following are full and zero length edges are allowed.

In fact, the number of full StT topologies, $\#\text{Top}(N)$, on an $N$-terminal set is super exponential [17]:

$$\#\text{Top}(N) = \frac{(2N - 4)!}{(N - 2)!2^{N - 2}} = (2N - 5)(2N - 7)\ldots5.3.1.$$  

For the heuristics presented here, we will take the approach of choosing a topology, then attempting to position the Steiner points in such a way as to give the shortest length network possible for that topology, which is exactly where the angles between the incident edges are $120^\circ$. In the Euclidean plane, there is a polynomial time algorithm [20] that can exactly position the Steiner points, but this algorithm cannot be generalised to higher dimensions [38]. Instead, an iterative method will be described and used here.
9.2 Greedy trees based on MST algorithms

Smith and Shor [37] define the “greedy tree” of an N-terminal set as follows. 

**KGT:** Start with all the N terminals, regarded as a forest of N 1-node trees. Extend the current forest by adding the shortest possible line segment that causes two trees to merge. Continue until the forest is completely merged into a single tree.

This greedy tree algorithm is a generalisation of Kruskal’s algorithm for a MST [27]. One of its stated properties is that the greedy tree it produces is no longer than the MST.

It is also claimed that another kind of greedy tree, which generalises Prim’s algorithm [34], is no longer than the MST. This greedy tree is defined as follows: 

**PGT:** Start with the line segment defined by the closest pair of two terminals, regarded as a 2-node, 1-edge tree. Extend the current tree by adding the shortest possible line segment which joins the current tree to an as-yet unattached terminal. Continue in this manner until all terminals are joined.

These greedy trees are StTs, with the Steiner points being the junctions created when a new line segment meets an existing one not at a terminal. The Steiner points have the property that two of the adjacent edges form a straight line and the third edge meets this line at 90°. A shorter tree is obtainable by moving each Steiner point till the angles are 120°. In fact, any movement of one, some or all of the Steiner points, one at a time towards the position where its incident angles are 120° will monotonically decrease the length of the tree. That is to say that although we cannot directly solve the system for the optimum configuration, we can iteratively move towards it, as will be described in the next section.

The Steiner points could be repositioned as a final step to these heuristics, or they could repositioned each time an edge is added, or repositioning could be used to determine what might be the best places to greedily add edges.
9.3 Methods for repositioning Steiner points

The length function of an StT is strictly convex [17] so if the Steiner points of a StT, $S$, with fixed topology are moved in such a way that the length is always decreasing, we can assume that $S$ is approaching its unique, minimum length configuration.

When the edges incident with a Steiner point, $s$, meet at $120^\circ$ the sum, $\vec{V}(s)$, of the unit vectors along the edges is zero. One method of repositioning is to move the Steiner points so that $\vec{V}$ decreases or becomes zero. Warren D. Smith [38] gives a generally convergent iteration which adjusts all the Steiner points simultaneously, until they reach optimal position. A simpler method is to move the Steiner points one at a time. There are several different ways to do this as described by various authors [29]. An outline of the method used here is as follows.

Suppose the Steiner points of the StT are in some order, and for a Steiner point $s$, let $\bar{L}(s)$ be the sum of the lengths of the three edges incident with $s$.

Choose a starting step size $\epsilon$

Choose a minimum for the step size, $\epsilon_0$

While ($\epsilon > \epsilon_0$)

For each junction, $s$

Measure $\bar{L}(s)$

Calculate $\vec{V}(s)$

Let $\vec{U}(s) = \vec{V}(s)/||\vec{V}(s)||$

Let $s' = s + \epsilon \vec{U}(s)$

Measure $\bar{L}(s')$

If $\bar{L}(s') < \bar{L}(s)$

replace $s$ with $s'$

if total tree length has not decreased

decrease $\epsilon$

End while
The direction of $\vec{V}$ is, in fact, the direction of steepest descent of the length function. The problems of detecting convergence and of avoiding a very long run time are circumvented here, by satisfying ourselves with an approximate solution and terminating the algorithm when small perturbations no longer shorten the tree. In practice, the algorithm terminates after a polynomial number of steps, and can also be set to terminate after a fixed number of steps.

9.4 New heuristics

We wish to establish conditions under which our new heuristics product StTs no longer than the MST. First we will define a fairly general operation which will be used iteratively to build a StT. Next we describe a condition on this operation which will give us our result.

9.4.1 Joining operations

**Definition 9.1** A joining operation (JoinOp) on a forest or a set of terminals (considered as a forest of 1-node trees) consists of the following two steps:

**Step 1:** Join two separate components of the forest by adding an edge. Each endpoint of the new edge may either be a terminal, if that terminal is isolated, or an existing edge $e$. If the latter, $e$ is split into two new edges (one or both of which may have length 0) creating a vertex at the junction. Call this new vertex a ‘junction’.

**Step 2:** Reposition all the junctions in the forest to reduce the total length of the forest (any junction is not counted as a terminal of the original set). For example, use the repositioning method of Section 9.3.

The cost, $C(f)$, of a JoinOp, $f$, is the difference in the length of the forest before and after the operation.
Let $P$ be a set of $N$ terminals.

**Lemma 9.2** After $(N - 1)$ joining operations are applied to $P$, the resulting graph $T_P$, is an StT with the junctions as Steiner points.

**Proof.** To be an StT, $T_P$ needs to be connected and a tree, with $N - 2$ Steiner points of degree 3, and all terminals of $P$ being leaves. Firstly, note from Step 1 that the degree of each junction is 3, and that each terminal is a leaf, since it can only get an edge attached to it if is isolated at the time. Moreover, no cycles are created, so $T_P$ is a tree.

Secondly, let $C_k$ be the number of connected components (including trivial components) of the forest on $P$ after $k$ JoinOps have been performed. Then $C_0 = N$ and, as each JoinOp decreases $C_k$ by 1, we see by induction that

\[ C_k = N - k \quad \text{and} \quad C_{N-1} = 1. \tag{9.1} \]

Thus $T_P$ is connected.

Finally, let $M$ be the number of junctions in $T_P$. Since $T_P$ is a tree it has $M + N - 1$ edges, and, since the sum of degrees in a graph equals twice the number of edges

\[ N + 3M = 2M + 2N - 2, \]

\[ M = N - 2. \]

Thus, with the junctions regarded as Steiner points, $T_P$ is an StT. \qed

### 9.4.2 Edge short joining operations

For a tree $T$ (or a single edge) let $L(T)$ denote the sum of the lengths of all the edges in $T$, where length is determined by the appropriate Euclidean metric $d(\ , \ )$ (this could be generalised to other metrics).

**Definition 9.3** Call a joining operation, $f$, which connects two components $T_a$ and $T_b$ of a forest an edge short joining operation (ES JoinOp) if its cost is no more than the
shortest distance from a terminal in \( T_a \) (or \( T_b \)) to a terminal in a different component:

\[
C(f) \leq \max\{ \min_{t \in A, s \in A} d(s, t), \min_{t \in B, s \in B} d(s, t) \},
\]

where \( A \) and \( B \) are the terminal sets of \( T_a \) and \( T_b \) respectively.

**Theorem 9.4** An StT built by performing \((N - 1)\) ES JoinOps on \( P \) is no longer than an MST on \( P \).

**Proof.** Let \( F_k, 1 \leq k \leq N \) be the forest on \( P \) after \((k - 1)\) ES JoinOps. Then \( F_N \) is a tree (StT) connecting all of \( P \), and \( \{F_k\} \) is a sequence of forests. Let \( M = M_1 \) be an MST on \( P \). \( M_k \) will be defined as \( M \) with a certain \( k - 1 \) edges removed, such that \( M_k \cup F_k \) is a tree connecting all of \( P \) (that is, an StT on \( P \)). We will show that we can choose which edge of \( M_k \) to remove so that the sequence of StTs, \( \{M_{k+1} \cup F_{k+1}\} \), is of non increasing length.

\[
M_1 \cup F_1 = M \quad \text{and so} \quad L(M_1 \cup F_1) \leq L(M).
\]

Assume that

\[
L(M_k \cup F_k) \leq L(M). \quad (9.2)
\]

The \( k \)th JoinOp joins two components, say \( T_1 \) and \( T_2 \) of \( F_k \). In \( M_k \cup F_k \) there is a unique path, \( p \), connecting \( T_1 \) to \( T_2 \), which contains at least one edge of \( M_k \). In particular there is an edge, \( e_1 \), of \( M_k \cap p \) with exactly one endpoint in \( T_1 \), and there is an edge, \( e_2 \), with exactly one endpoint in \( T_2 \) (these may be the same edge). Because the \( k \)th JoinOp is an ES JoinOp its cost, \( C(k) \), is no greater than the length of at least one of these edges, say \( e_1 \):

\[
C(k) \leq L(e_1). \quad (9.3)
\]
Let $M_{k+1} = M_k \setminus e_1$. Then $M_{k+1} \cup F_k + 1$ is again an StT and

$$L(M_{k+1} \cup F_k + 1) = L(F_k + 1) + L(M_{k+1})$$

$$= L(F_k) + C(k) + L(M_k) - L(e_1)$$

$$\leq L(F_k) + L(M_k)$$

$$= L(M_k \cup F_k)$$

$$\leq L(M),$$

by equations (9.2) and (9.3). So, by induction

$$L(F_N) = L(F_N \cup M_N) \leq L(M).$$

This result holds regardless of the size of the error in our repositioning iteration, so long as the condition of Definition 9.3 is satisfied.

### 9.4.3 New Kruskal and Prim type greedy heuristics

Now we describe specific heuristics for StTs:

**KGS:** Following Kruskal’s MST algorithm and KGT, let KGS($P$) (a Kruskal Greedy Steiner tree on $P$) be the StT that results when the JoinOp with minimum cost over all available JoinOps is chosen each time.

In a Prim type algorithm we start by joining two terminals of $P$ to make a 2-node tree, then at each subsequent stage we join another terminal to the current tree. This induces an ordering of the terminals of $P$, which is the order in which they are joined to the current tree.

Let $\{p_k\}, 1 \leq k \leq N$, be an ordering induced on $P$ by Prim’s MST algorithm; $p_1$ and $p_2$ are two terminals of $P$ closest together, and, when the current tree $T$ spans $k - 1$ terminals, $p_k$ is the terminal which can be joined to $T$ using the shortest edge from a terminal in $T$. 
**PGS1:** Let $\text{PGS1}(P)$ be the StT that results when we join terminals to the current tree in the order $\{p_k\}$, each time using the least costly JoinOp available.

**PGS2:** Again following Prim’s algorithm and PGT, let $\text{PGS2}(P)$ be the StT that results when we start by joining the two terminals that are closest together, then each time joining another terminal to the current tree by the least costly JoinOp available.

**Corollary 9.5** $\text{KGS}(P)$, $\text{PGS1}(P)$ and $\text{PGS2}(P)$ are each no longer than $\text{MST}(P)$.

**Proof.** Firstly, note that for any component $T_a$ of a forest there exists an ES JoinOp which can be used to connect $T_a$ to another component of the forest (add the shortest edge from a terminal in $T_a$ to a terminal not in $T_a$).

For $\text{KGS}(P)$ we choose the least costly JoinOp overall so it must be an ES JoinOp.

In the case of $\text{PGS1}(P)$, $T_a$ is the current tree and $T_b$ is the terminal to be added following Prim’s MST algorithm. Thus the distance $d$ from $T_b$ to the nearest terminal in $T_a$ is no longer than the distance from any terminal in $T_a$ to any other not in $T_a$. We choose the least costly JoinOp joining $T_a$ and $T_b$, which must be an ES JoinOp.

Finally, for $\text{PGS2}(P)$, $T_a$ is again the current tree and we choose the least costly JoinOp which adds another terminal to $T_a$. This also must be an ES JoinOp. The result follows by Theorem 9.4.

In the same way, we can use Theorem 9.4 to show that the outputs of KGT and PGT are each no longer than an MST.

Of course, this method of proof cannot be used to compare any two StTs; the cost of adding some edge $e$ may be less than that of adding a different edge $f$ at some time, but this does not guarantee that at some later time removing $e$ and adding $f$ will not result in a shorter StT. The choice of available JoinOps and their costs is dependent on the structure of the tree “so far” and the order in which they are performed; we cannot compare JoinOps at different stages of construction. (Of course, if it were possible to know which choice of
joining at any stage would give the shortest StT in the end, we would have a polynomial algorithm for the Minimal Steiner Tree, and a demonstration that $P = NP$!

This proof shows that we do not need to be totally greedy, selecting the least cost operation in each step, to guarantee a result at least as good as an MST. We could define ‘quasi greedy tree’ heuristics or randomised heuristics, where at each stage we choose some ES JoinOp according to different criteria.

9.4.4 A metropolis algorithm for the EST problem

We compare the performance of our new heuristics to the MST and also to that of a metropolis algorithm, MA, for the EST problem.

MA generates an initial Steiner tree topology, $S(0)$, and tries a random perturbation at each step. The perturbed tree, $S'$, is generated by removing a randomly selected edge of the current tree, $S(t)$, and rejoining the two subtrees with an available JoinOp chosen at random. If the length of $S'$ is less than that of $S(t)$ then $S(t + 1) = S'$, otherwise $S(t + 1) = S'$ with some small probability that is a function of the difference in length between $S(t)$ and $S'$. Otherwise, $S(t + 1) = S(t)$. MA stops at a fixed time $T_f$, and outputs the Steiner tree $S(t_m)$, $0 \leq t_m \leq T_f$ of shortest length.

Lundy and Mees [29] have previously applied a similar simulated annealing algorithm to the Steiner tree problem. We dispensed with the ‘cooling schedule’ typically used in simulated annealing, and opted for a more radical perturbation scheme than Lundy and Mees, who, in their algorithm, removed and rejoined a randomly selected terminal of the tree in each step. We found that, after some adjusting of parameters, this scheme converged to more rapidly to the near optimal parts of the state space, and moved more freely around the near optimal solutions. We set MA to output the best solution that it finds along the way, rather than the final solution, so there was no advantage in applying a cooling schedule to try and force the algorithm to finish at the best solution.
9.5 Some results and conclusions

The heuristics KGS, PGS1 and PGS2 were compared to KGT, PGT and the MST on sets of 10 and 50 terminals uniformly distributed on \((0, 1) \times (0, 1) \times (0, 1)^N\). Two additional deterministic heuristics, KGTopt and PGTopt are also considered. These are, respectively, KGT and PGT with the Steiner point repositioning algorithm applied to the final tree.

Table 9.1: StT length compared to MST. \(N = 10, 50\) trials.

<table>
<thead>
<tr>
<th></th>
<th>MST</th>
<th>MA</th>
<th>KGT</th>
<th>KGTopt</th>
<th>PGT</th>
<th>PGTopt</th>
<th>KGS</th>
<th>PGS1</th>
<th>PGS2</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>2.266</td>
<td>0.129</td>
<td>0.000</td>
<td>1.285</td>
<td>0.000</td>
<td>1.513</td>
<td>1.513</td>
<td>1.513</td>
<td></td>
</tr>
<tr>
<td>10th perc.</td>
<td>2.890</td>
<td>3.109</td>
<td>0.133</td>
<td>2.438</td>
<td>0.121</td>
<td>2.590</td>
<td>2.590</td>
<td>2.589</td>
<td></td>
</tr>
<tr>
<td>mean</td>
<td>3.248</td>
<td>4.873</td>
<td>0.805</td>
<td>4.514</td>
<td>0.705</td>
<td>4.717</td>
<td>4.597</td>
<td>4.652</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.2: StT length compared to MST. \(N = 50, 19\) trials.

<table>
<thead>
<tr>
<th></th>
<th>MST</th>
<th>MA</th>
<th>KGT</th>
<th>KGTopt</th>
<th>KGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>min</td>
<td>8.676</td>
<td>3.879</td>
<td>0.536</td>
<td>3.497</td>
<td>3.804</td>
</tr>
<tr>
<td>10th perc.</td>
<td>9.046</td>
<td>4.307</td>
<td>0.639</td>
<td>3.629</td>
<td>4.184</td>
</tr>
<tr>
<td>mean</td>
<td>9.546</td>
<td>5.066</td>
<td>0.919</td>
<td>4.434</td>
<td>4.923</td>
</tr>
<tr>
<td>90th perc.</td>
<td>10.191</td>
<td>6.141</td>
<td>1.194</td>
<td>4.988</td>
<td>5.505</td>
</tr>
<tr>
<td>max</td>
<td>10.389</td>
<td>6.236</td>
<td>1.750</td>
<td>5.088</td>
<td>6.167</td>
</tr>
</tbody>
</table>

In Tables 1 and 2, the average length of the MST and the average performance of each heuristic, in terms of the percent difference (decrease) in tree length compared to the MST, is given. The mean, maximum, minimum, 10th and 90th percentile are given for
each data set.

Table 9.3: StT length compared to MA. $N = 10, 50$ trials.

<table>
<thead>
<tr>
<th></th>
<th>MA</th>
<th>MST</th>
<th>KGT</th>
<th>KGTopt</th>
<th>PGT</th>
<th>PGTopt</th>
<th>KGS</th>
<th>PGS1</th>
<th>PGS2</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90th perc.</td>
<td>3.487</td>
<td>7.273</td>
<td>5.810</td>
<td>1.417</td>
<td>5.860</td>
<td>1.693</td>
<td>0.964</td>
<td>1.093</td>
<td>1.000</td>
</tr>
<tr>
<td>mean</td>
<td>3.088</td>
<td>5.157</td>
<td>4.302</td>
<td>0.382</td>
<td>4.407</td>
<td>0.538</td>
<td>0.168</td>
<td>0.295</td>
<td>0.236</td>
</tr>
<tr>
<td>10th perc.</td>
<td>2.748</td>
<td>3.209</td>
<td>2.612</td>
<td>0.000</td>
<td>2.679</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>min</td>
<td>2.212</td>
<td>0.129</td>
<td>-0.024</td>
<td>-2.478</td>
<td>-0.024</td>
<td>-2.478</td>
<td>-2.477</td>
<td>-2.476</td>
<td>-2.476</td>
</tr>
</tbody>
</table>

Table 9.4: StT length compared to MA. $N = 50, 19$ trials.

<table>
<thead>
<tr>
<th></th>
<th>MA</th>
<th>MST</th>
<th>KGT</th>
<th>KGTopt</th>
<th>KGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>9.753</td>
<td>6.651</td>
<td>5.355</td>
<td>1.569</td>
<td>0.846</td>
</tr>
<tr>
<td>90th perc.</td>
<td>9.662</td>
<td>6.542</td>
<td>5.114</td>
<td>1.291</td>
<td>0.551</td>
</tr>
<tr>
<td>mean</td>
<td>9.061</td>
<td>5.341</td>
<td>4.371</td>
<td>0.667</td>
<td>0.151</td>
</tr>
<tr>
<td>10th perc.</td>
<td>8.622</td>
<td>4.501</td>
<td>3.819</td>
<td>0.130</td>
<td>-0.180</td>
</tr>
<tr>
<td>min</td>
<td>8.233</td>
<td>4.036</td>
<td>3.478</td>
<td>0.043</td>
<td>-0.242</td>
</tr>
</tbody>
</table>

In Tables 3 and 4 the performance of the deterministic heuristics is compared to that of MA, by considering the percent difference (increase) in length of the MST and the output of each heuristic compared to the output of MA. The mean, maximum, minimum, 10th and 90th percentile of each data set are given.

There is very little actual MSTT data available for sets of terminals in Euclidean 3-space with which to compare the output of StT heuristics, however the output of MA (after suitable tweaking of parameters and runtime) seems to be very close to optimal, because repeated trials on the same data sets yield the same best StT in most instances. For this reason the output of MA serves as a perhaps more consistent point of comparison for
heuristic performance than MST length.

The results indicate that, on average, the heuristics which reposition Steiner points at each step (KGS, PGS1 and PGS2) perform better than those that reposition as a final step (KGTopt and PGTopt), and much better than those that don’t reposition at all (KGT and PGT), which suggests that repositioning is a powerful tool for both optimising an existing tree and choosing a good topology.

In fact, the output of KGS was on average less than 0.2% longer than the output of MA in both the 10 and 50 terminal trials, whilst the output of KGT was about 4.3% longer than MA, and KGTopt was 0.38% longer on 10 terminals and 0.67% longer on 50 terminals. Of the deterministic heuristics, KGS is the one with the maximum degree of choice in each step and does, on average, outperform the others, although, in comparison to PGS1 and PGS2 the difference is fairly marginal.

Not surprisingly, KGS also has the longest runtime of the deterministic heuristics, taking about 15 – 20 seconds for one trial on 50 terminals and just a couple of minutes for 100 trials on 10 terminals. Also, although MA has a much longer run time than KGS, we see that the resulting StT is less than 0.2% shorter than that found using KGS.

Figures 1, 2, 3 and 4 are plots of the data in Tables 1, 2, 3 and 4.
Figure 1: Percent decrease in length from MST
N=10, 50 trials

Figure 2: Percent decrease in length from MST
N=50, 19 trials
Figure 3: Percent increase in length from MA
N=10, 50 trials

Figure 4: Percent increase in length from MA
N=50, 19 trials
Bibliography


[38] W. D. SMITH, How To Find Steiner Minimal Trees in Euclidean $d$-Space *Algorithmica* 7 (1992), no. 2–3, 137–177.


Appendix A

Proof of Lemma 6.27

It needs to be shown that for all \( k \geq 7 \) and each \( w \in [k - 1] \)

\[
F(k, w, 2k) - D(k, 2k) > 0,
\]

with \( F(k, w, \mu) \) and \( D(k, \mu) \) defined by (6.29) and (6.33).

Define \( N(k, r) \) to be

\[
N(k, r) := \frac{(2k)!e^{2k}}{(2k)^{2k+1}} (F(k, k - r, 2k) - D(k, 2k)).
\]

Then, using (6.35),

\[
N(k, r) = \sum_{j=1}^{r-1} \frac{\lfloor 2k \rfloor_j}{(2k)^j + 1} + \frac{(2k)!f_{2k}(2k)}{(2k)^{2k}(r + 2)} + \frac{1}{r + 2} \sum_{j=1}^{r} \frac{\lfloor 2k \rfloor_j}{(2k)^j} + \frac{(2k)!}{(2k)^{2k}} \frac{f_{2k+1}(2k) - f_{2k}(2k)}{2f_{2k+1}(2k) - f_{2k}(2k)},
\]

where \([n]_i \) denotes \( n!/(n - i)! \). Clearly \( F(k, w, 2k) - D(k, 2k) \) is positive if and only if \( N(k, r) \) is positive.

It will be shown using a lower bound that \( N(k, r) \) is positive for \( k \geq 18 \) and each \( r \). For \( k = 7, \ldots, 17 \) and each \( r \), \( N(k, r) \) will be calculated directly.
Note that
\[
\frac{f_{2k+1}(2k) - f_{2k}(2k)}{2f_{2k+1}(2k) - f_{2k}(2k)} = \frac{-\frac{(2k)^2}{(2k)!}}{-2\frac{(2k)^2}{(2k)!} + f_{2k}(2k)} = \frac{-1}{\frac{(2k)!}{(2k)!} f_{2k}(2k) - 2}.
\]

So
\[
N(k, r) = \sum_{j=1}^{r-1} \frac{[2k]_j}{(2k)^{j+1}} + \frac{(2k)! f_{2k}(2k)}{(2k)^{2k(r + 2)}} + \frac{1}{(r + 2)} \sum_{j=1}^{r} \frac{[2k]_j}{(2k)^j} - \frac{\frac{(2k)!}{(2k)!} f_{2k}(2k)}{(2k)! f_{2k}(2k) - 2}.
\]

Let
\[
A(k) := \frac{(2k)!}{(2k)^{2k}} f_{2k}(2k),
\]
\[
B(k, r) := \sum_{j=1}^{r} \frac{[2k]_j}{(2k)^j},
\]
\[
C(k, r) := \sum_{j=1}^{r-1} \frac{j[2k]_j}{(2k)^{j+1}},
\]
so that
\[
N(k, r) = C(k, r) + \frac{A(k) + B(k, r)}{r + 2} - \frac{A(k)}{A(k) - 2}.
\]

Lower bounds for \(A(k), B(k, r)\) and \(C(k, r)\) will give a lower bound for \(N(k, r)\). First, consider \(A(k)\). By Stirling’s formula
\[
\frac{(2k)!}{(2k)^{2k}} > \sqrt{2\pi} 2^k e^{-2k}, \tag{A.1}
\]
and \(f_{2k}(2k)\) is a portion of the Poisson expansion of \(e^{2k}\):
\[
f_{2k}(2k) = \sum_{j \geq 2k} \frac{(2k)^j}{j!} = \delta e^{2k}, \tag{A.2}
\]
where \(0 < \delta < 1\). In fact \(\delta > \frac{1}{2}\) by the following claim:
Claim A.1 For any positive integer $m$

\[ f_m(m) > \frac{1}{2}e^m. \]

Proof. It will be shown that

\[ \sum_{i=m}^{2m-1} \frac{(m)^i}{i!} \geq \sum_{i=0}^{m-1} \frac{(m)^i}{i!}. \] (A.3)

Here the left hand side is less than $f_m(m)$, as the latter sums from $m$ to $\infty$, and the right hand side equals $e^m - f_m(m)$. Thus the claim follows.

Comparing terms in (A.3), for $j = 0, \ldots, m - 1$, let

\[ \frac{(m)^{m+j}}{(m+j)!} = \beta_j \frac{(m)^{m-j-1}}{(m-j-1)!}. \]

Then

\[ \beta_j = \frac{(m)^{2j+1}}{(m+j)_{2j+1}} \]
\[ = \frac{m^j}{m} \prod_{i=1}^{m} \frac{m}{m-i} \frac{m+i}{m} \]
\[ = \prod_{i=1}^{j} \frac{m^2}{m^2 - i^2} \]
\[ \geq 1. \]

Therefore, for each $j = 0, \ldots, m - 1$,

\[ \frac{(m)^{m+j}}{(m+j)!} \geq \frac{(m)^{m-j-1}}{(m-j-1)!}, \]

which proves (A.3) \qed

Therefore, multiplying (A.1) by (A.2) and substituting $\frac{1}{2}$ for $\delta$,

\[ A(k) > \sqrt{\pi k}. \]

In what follows, it will be convenient to use the substitutions

\[ \rho = \frac{r - 1}{\sqrt{2k}} \]
\[ m = \sqrt{2k}. \]

Let
\[ \tilde{A}(m) := \sqrt{\frac{\pi}{2m}}. \]  \hfill (A.4)

Then \( \tilde{A}(m) \) is increasing with \( m \), and for all \( k \), \( A(k) > \tilde{A}(\sqrt{2k}) \).

For \( B(k, r) \),
\[
B(k, r) = \sum_{i=1}^{r} \frac{\lfloor 2k \rfloor_i}{(2k)^i}
= \sum_{i=1}^{r} \prod_{t=0}^{i-1} \frac{2k-t}{2k}
= \sum_{i=1}^{r} \exp \left( \sum_{t=0}^{i-1} \ln \left(1 - \frac{t}{2k}\right) \right)
= \sum_{i=1}^{\min\{r, \lfloor m \rfloor - 1\}} \exp \left( \sum_{t=0}^{i-1} \ln \left(1 - \frac{t}{2k}\right) \right) + \sum_{i=\lfloor m \rfloor}^{r} \exp \left( \sum_{t=0}^{i-1} \ln \left(1 - \frac{t}{2k}\right) \right).
\]

Let
\[ B_1(k, r) := \sum_{i=1}^{\min\{r, \lfloor m \rfloor - 1\}} \exp \left( \sum_{t=0}^{i-1} \ln \left(1 - \frac{t}{2k}\right) \right) \]
and
\[ B_2(k, r) := \sum_{i=\lfloor m \rfloor}^{r} \exp \left( \sum_{t=0}^{i-1} \ln \left(1 - \frac{t}{2k}\right) \right). \]

In the range \( 0 \leq x \leq x_0 < 1 \), \( \ln(1 - x) \) is convex and is lower bounded by the line from the origin to \((x_0, \ln(1 - x_0))\), that is, the line \( y(x) = \frac{\ln(1-x_0)}{x_0} x \).

Hence, for \( \frac{t}{2k} < x_0 \),
\[ \ln \left(1 - \frac{t}{2k}\right) > \frac{\ln(1-x_0)}{x_0} \frac{t}{2k}. \]

For, \( i \leq \lfloor m \rfloor \) and \( t \leq i - 1 \), we have \( \frac{t}{2k} < \frac{m}{2k} = \frac{1}{m} \), so
\[
\sum_{t=0}^{i-1} \ln \left(1 - \frac{t}{2k}\right) > m \ln \left(1 - \frac{1}{m}\right) \sum_{t=0}^{i-1} \frac{t}{m^2}
= \frac{i(i-1)}{2m} \ln \left (1 - \frac{1}{m}\right). \quad \text{(A.5)}
\]
For $[m] < i < k$, we have $\frac{i-1}{2k} < \frac{k-1}{2k} < \frac{1}{2}$, so
\[
\sum_{t=[m]}^{i-1} \ln(1 - \frac{t}{2k}) > 2 \ln \left( \frac{1}{2} \right) \sum_{t=[m]}^{i-1} \frac{t}{m^2}
= 2 \ln \left( \frac{1}{2} \right) \left( \frac{i(i-1) - [m][[m] - 1]}{2m^2} \right).
\tag{A.6}
\]

Hence, for $r \leq [m] - 1$,
\[
B_1(k, r) > \sum_{i=1}^{r} \exp \left( \frac{i(i-1)}{2m} \ln \left( 1 - \frac{1}{m} \right) \right)
= \sum_{i=1}^{r} \left( \frac{1}{m} \right)^{i(i-1) \frac{2m}{m^2}}
= \sum_{i=1}^{r} \left( \frac{m}{m-1} \right)^{-\frac{i(i-1)}{2m}}
> \int_{1}^{r+1} \left( \frac{m}{m-1} \right)^{-\frac{u(u-1)}{2m}} du,
\]
since for constant $K > 1$ and constant $c > 0$, $K^{-u(u-1)/c}$ is decreasing for $u \geq 1$. Substituting $\rho = (r - 1)/m$ and $u = \frac{\rho}{m}$, so that $dv = m\,du$, gives
\[
B_1(k, r) > \int_{\frac{1}{m}}^{\frac{1}{m}+\frac{2}{m}} m \left( \frac{m}{m-1} \right)^{-\frac{u(u-1)}{2m}} du.
\]

Now we show that for $0 \leq u \leq 1$,
\[
\left( \frac{t}{t-1} \right)^{\frac{-u}{2}(tu-1)} \geq 1 - \frac{u}{2}.
\tag{A.7}
\]
In the following we use that $\ln(1 - x) \leq -x$ for $0 \leq x \leq 1$, and $\ln(1 + x) \leq x$ for $x \geq 0$.

Taking the log of both sides gives
\[
\frac{-u}{2}(tu-1) \ln \left( \frac{t}{t-1} \right) \geq \ln(1 - \frac{u}{2}),
\]
which is implied by
\[
\frac{-u}{2}(tu-1) \ln \left( \frac{t}{t-1} \right) \geq \frac{-u}{2}
\]
\[
(tu-1) \ln \left( 1 + \frac{1}{t-1} \right) \leq 1,
\]
197
which follows if

\[(tu - 1)\left(\frac{1}{t - 1}\right) \leq 1.\]

This holds for all \(u \leq 1\).

Hence, with \(\rho^* := \min\{\rho + \frac{2}{m}, 1\},\)

\[
\int_{\frac{1}{m}}^{\rho + \frac{2}{m}} \left(\frac{m}{m - 1}\right)^{\rho\left(mu + 1\right)} du > \int_{\frac{1}{m}}^{\rho^*} \left(1 - \frac{u}{2}\right) du
\]

\[
= \left(\rho^* - \frac{\rho^2}{4} - \frac{1}{m} + \frac{1}{4m^2}\right).
\]

Now, \(\rho + \frac{2}{m} \geq 1\) if and only if \(r + 1 \geq m\), and \(m - 1 \leq r \leq [m] - 1\) if and only if \(r = [m] - 1\).

So, for \(r \leq [m] - 2\), \(\rho^* = \rho + \frac{2}{m}\) and for \(r = [m] - 1\), \(\rho^* = 1\).

Hence for \(r \leq [m] - 2\),

\[
B_1(k, r) > m \left(\left(\rho + \frac{2}{m}\right) - \frac{(\rho + \frac{2}{m})^2}{2} - \frac{1}{m} + \frac{1}{4m^2}\right)
\]

\[
= m \left(\rho(1 - \frac{\rho}{4}) + \frac{1}{m} \left(1 - \rho - \frac{3}{4m}\right)\right)
\]

\[
> m \left(\rho(1 - \frac{\rho}{4}) + \frac{1}{m} \left(\frac{2}{m} - \frac{3}{4m}\right)\right)
\]

\[
= m \left(\rho(1 - \frac{\rho}{4}) + \frac{5}{4m^2}\right)
\]

\[
> m\rho \left(1 - \frac{\rho}{4}\right),\quad \text{(A.8)}
\]

which is increasing with \(m\) and therefore \(k\).

For \(r = [m] - 1\),

\[
B_1(k, [m] - 1) > m \left(1 - \frac{1}{4} - \frac{1}{m} + \frac{1}{4m^2}\right)
\]

\[
= m \left(\frac{3}{4} - \frac{1}{m} + \frac{1}{4m^2}\right),\quad \text{(A.9)}
\]

which is also increasing with \(m\) and therefore \(k\).
For \( r \geq \lceil m \rceil \),
\[
B(k, r) = B(k, \lceil m \rceil - 1) + B_2(k, r)
\]
and
\[
B_2(k, r) = \sum_{i=\lceil m \rceil}^{r} \exp \left( \sum_{t=0}^{[m]-1} \ln(1 - \frac{t}{2k}) + \sum_{t=\lceil m \rceil}^{i-1} \ln(1 - \frac{t}{2k}) \right)
\]
\[
> \sum_{i=\lceil m \rceil}^{r} \exp \left( \frac{[m]([m] - 1)}{2m} \ln \left( 1 - \frac{1}{m} \right) + 2 \ln \left( \frac{1}{2} \right) \right) \left( \frac{i(i-1) - [m][[m] - 1]}{2m^2} \right),
\]
using (A.5) and (A.6). Hence
\[
B_2(k, r) > \sum_{i=\lceil m \rceil}^{r} \exp \left( \frac{[m]([m] - 1)}{2m} \ln \left( 1 - \frac{1}{m} \right) + 2 \ln \left( \frac{1}{2} \right) \right) \left( \frac{i(i-1) - [m][[m] - 1]}{2m^2} \right),
\]
since \( \ln \left( \frac{4^{\frac{1}{m} (m-1)}}{m} \right) \) is non-negative for all \( k \geq 2 \). Thus
\[
B_2(k, r) > \sum_{i=\lceil m \rceil}^{r} \exp \left( \ln \left( \frac{4^{\frac{1}{m} (m-1)}}{m} \right) \right) + \ln \left( \frac{2}{\frac{(i-1)}{m^2}} \right)
\]
\[
= \left( \frac{4^{\frac{1}{m} (m-1)}}{m} \right) \sum_{i=\lceil m \rceil}^{r} 2^{-\frac{(i-1)}{m^2}}
\]
\[
= 2^{\frac{m-1}{m}} \left( \frac{m - 1}{m} \right)^{\frac{m-1}{2}} \sum_{i=\lceil m \rceil}^{r} 2^{-\frac{(i-1)}{m^2}}.
\]
The terms in the summation are decreasing with \( i \), so
\[
B_2(k, r) > 2^{\frac{m-1}{m}} \left( \frac{m - 1}{m} \right)^{\frac{m-1}{2}} \int_{\lceil m \rceil}^{r+1} 2^{-\frac{u(u-1)}{m^2}} du.
\]
On substituting \( u = \frac{\rho}{m} \) and \( \rho = \frac{r+1}{m} \),
\[
\int_{\lceil m \rceil}^{r+1} 2^{-\frac{u(u-1)}{m^2}} du = m \int_{\lceil m \rceil}^{\frac{r+2}{m}} 2^{-u(u-\frac{1}{m})} du
\]
\[
= m \int_{1+\frac{1}{m}}^{\frac{r+2}{m}} 2^{-u(u-\frac{1}{m})} du
\]
\[
= m \int_{1+\frac{1}{m}}^{\frac{r+2}{m}} 2^{-u\rho} du,
\]
199
which is increasing with \(m\), as is \(2^{m-1} \left(\frac{m-1}{m}\right)^{m-1}\). Hence, for \(r \geq \lceil m \rceil\),

\[
B_2(k, r) > 2^{m-1} \left(\frac{m-1}{m}\right)^{m-1} m \left[\frac{-2^{-\rho} \ln(2)}{\rho \ln(2)}\right]^{\frac{1}{1+\frac{1}{m}}} m^{-1} \frac{1}{m} \left(2^{\rho(1+\frac{1}{m})} - 2^{-\rho(\rho+\frac{1}{m})}\right),
\]

(A.10)

where \(m = \sqrt{2k}\) and \(\rho = \frac{r-1}{m}\). This lower bound is increasing with \(m\) and therefore \(k\).

Putting together (A.8), (A.9) and (A.10), let

\[
\tilde{B}_{1a}(m, \rho) := m \rho \left(1 - \frac{\rho}{4}\right),
\]

\[
\tilde{B}_{1b}(m) := m \left(\frac{3}{4} - \frac{1}{m} + \frac{1}{m^2}\right),
\]

\[
\tilde{B}_2(m, \rho) := \tilde{B}_{1b}(m) + m \frac{2^{m-1}}{\rho \ln(2)} \left(\frac{m-1}{m}\right)^{\frac{1}{2}} \left(2^{-\rho(1+\frac{1}{m})} - 2^{-\rho(\rho+\frac{1}{m})}\right),
\]

and let

\[
\tilde{B}(m, \rho) := \begin{cases} 
\tilde{B}_{1a}(m, \rho) & \text{for } \rho \leq \frac{\lceil m \rceil - 2}{m}, \\
\tilde{B}_{1b}(m) & \text{for } \rho = \frac{\lceil m \rceil - 1}{m}, \\
\tilde{B}_2(m, \rho) & \text{for } \rho \geq \frac{\lceil m \rceil}{m}.
\end{cases}
\]

(A.11)

Then, as observed above, \(\tilde{B}(m, \rho)\) is increasing with \(m\), and for all \(k\) and \(r \in [k-1],\)

\[
B(k, r) > \tilde{B}(\sqrt{2k}, \frac{r-1}{\sqrt{2k}}).
\]

Now, consider \(C(k, r)\).

\[
C(k, r) = \sum_{j=1}^{r-1} h_j(2k),
\]

where \(h_j(n) = \frac{j \lfloor n/j \rfloor}{n}\). The series \(C(k, r)\) has the following two properties.

**Claim A.2** The sequence \(\{h_j(n)\}\) is unimodal:

\[
h_{j-1}(n) < h_j(n) \text{ if and only if } j(j-1) < n,
\]

and \(h_{j-1}(n) = h_j(n)\) if \(j(j-1) = n\).
Proof. The following are equivalent:

\[ h_{j-1}(n) < h_j(n), \]
\[ \frac{j-1}{n} \frac{[n]_{j-1}}{n^{j-1}} < \frac{j}{n} \frac{[n]_j}{n^j}, \]
\[ (j-1)n < j(n-j+1), \]
\[ -n < -j^2 + j, \]
\[ j(j-1) < n. \]

Clearly if \( j(j-1) = n \) then equality holds.

Claim A.3

\[ \sum_{j=1}^{n} h_j(n) = 1. \]

Proof.

\[
\sum_{j=1}^{n} h_j(n) = \sum_{j=0}^{n} \frac{jn!}{(n-j)!n^{j+1}}
\]
\[
= \frac{n!}{n^n} \sum_{j=0}^{n} \frac{jn^{n-j-1}}{(n-j)!}
\]
\[
= \frac{n!}{n^n} \sum_{t=0}^{n} \frac{(n-t)n^{t-1}}{t!}
\]
\[
= \frac{n!}{n^n} \left( \sum_{t=0}^{n} \left( \frac{n^t}{t!} - \frac{tn^{t-1}}{t!} \right) \right)
\]
\[
= \frac{n!}{n^n} \left( \sum_{t=0}^{n} \frac{n^t}{t!} - \frac{n^{t-1}}{t-1}! \right)
\]
\[
= \frac{n!}{n^n} \left( \sum_{t=0}^{n} \frac{n^t}{t!} - \sum_{t=0}^{n-1} \frac{n^t}{t!} \right)
\]
\[
= \frac{n!}{n^n} \left( \frac{n^n}{n!} - \sum_{t=0}^{n-1} \frac{n^t}{t!} \right)
\]
\[
= 1. \]

Let \( \gamma(k) \) be the positive solution of \( \gamma(\gamma - 1) = 2k \). Then
\[
\gamma(k) = \sqrt{2k + \frac{1}{4}} + \frac{1}{2} > \sqrt{2k}.
\]
Now

\[ C(k, r) = \min\{r-1, [m]-1\} \sum_{j=1}^{r-1} \frac{j \, [2k]_j}{2k (2k)^j} + \sum_{j=[m]}^{r-1} \frac{j \, [2k]_j}{2k (2k)^j}. \]

Let

\[ C_1(k, r) := \min\{r-1, [m]-1\} \sum_{j=1}^{r-1} \frac{j \, [2k]_j}{2k (2k)^j}, \]

and let

\[ C_2(k, r) := \sum_{j=[m]}^{r-1} \frac{j \, [2k]_j}{2k (2k)^j}. \]

In \( C_1(k, r) \), \( j \leq [m] - 1 \leq [\gamma(k)] - 1 \). Hence \( j < \gamma(k) \) and by Claim A.2 the terms in the series are monotonically increasing. This means \( C_1(k, r) \) can be lower bounded by an integral from 0 to \( r-1 \), as is done in the following.

First assume that \( r - 1 \leq [m] - 1 \). Then

\[ C_1(k, r) = \sum_{j=1}^{r-1} \frac{j \, [2k]_j}{2k (2k)^j} \]

\[ = \sum_{j=1}^{r-1} \frac{j}{2k} \prod_{t=0}^{j-1} \frac{2k-t}{2k} \]

\[ = \sum_{j=1}^{r-1} \frac{j}{2k} \exp\left( \sum_{t=0}^{j-1} \ln\left(1 - \frac{t}{2k}\right)\right). \]

Now, \( t \leq r - 2 < [m] - 2 < m \) so using (A.5) and substituting \( m \) for \( \sqrt{2k} \),

\[ C_1(k, r) > \sum_{j=1}^{r-1} \frac{j}{m^2} \exp\left( \frac{j(j-1)}{2m} \ln\left(1 - \frac{1}{m}\right)\right) \]

\[ = \sum_{j=1}^{r-1} \frac{j}{m^2} \left( \frac{m}{m-1} \right)^{-\frac{j(j-1)}{2m}} \]

\[ > \int_0^{r-1} \frac{v}{m^2} \left( \frac{m}{m-1} \right)^{-\frac{v(v-1)}{2m}} dv \]

\[ = \int_0^\rho u \left( \frac{m}{m-1} \right)^{-\frac{u(u-1)}{2}} du, \]
on substituting \( u = v/m \) and \( \rho = \frac{r-1}{m} \). Now \( \rho \leq \frac{\lceil m \rceil - 1}{m} < 1 \) so by (A.7),

\[
C_1(k, r) > \int_0^\rho u \left( 1 - \frac{u}{2} \right) du \\
= \left[ \frac{u^2}{2} - \frac{u^3}{6} \right]_0^\rho \\
= \frac{\rho^2}{2} - \frac{\rho^3}{6},
\]

for \( r \leq \lceil m \rceil \), and \( \rho \leq \frac{\lceil m \rceil - 1}{m} \). For \( r = \lceil m \rceil \), \( \rho = \frac{\lceil m \rceil - 1}{m} \leq 1 - \frac{1}{m} \), hence for \( r \leq \lceil m \rceil - 1 \),

\[
C_1(k, r) > \frac{\rho^2}{2} - \frac{\rho^3}{6}, \quad (A.12)
\]

and for \( r = \lceil m \rceil \),

\[
C_1(k, \lceil m \rceil) > \frac{1}{2} (1 - \frac{1}{m})^2 - \frac{1}{6} (1 - \frac{1}{m})^3. \quad (A.13)
\]

Now for \( C_2(k, r) \),

\[
C_2(k, r) = \sum_{j=\lceil m \rceil}^{r-1} \frac{j}{2k} \exp \left( \sum_{t=1}^{j-1} \ln \left( 1 - \frac{t}{2k} \right) \right).
\]

Using the same lower bounds as for \( B_2(k, r) \), and substituting \( m \) for \( \sqrt{2k} \) gives

\[
C_2(k, r) > 2^{\frac{m-1}{m}} \left( \frac{m-1}{m} \right)^{\frac{m-1}{2}} \sum_{j=\lceil m \rceil}^{r-1} \frac{j}{m^2} 2^{-\frac{j(j-1)}{m^2}}.
\]

Let \( f(v) = v^{\frac{-v(v-1)}{m^2}} \). Then

\[
f'(v) = 2^{\frac{-v(v-1)}{m^2}} \left( 1 - \frac{2v^2 - v}{m^2} \ln(2) \right),
\]

and \( f'(v) < 0 \) if

\[
\frac{2v^2 - v}{m^2} \ln(2) > 1,
\]

which holds for all \( v > m \) provided

\[
2m^2 - m > \frac{m^2}{\ln(2)} \\
2 - \frac{1}{m} > \frac{1}{\ln(2)} \\
m > \frac{1}{2 - \frac{1}{\ln(2)}}.
\]
The latter holds for all $m \geq 2$, or equivalently, $k \geq 2$.

Thus the terms in the summation are decreasing and

$$C_2(k, r) > 2 \sum_{j=r}^{m-1} h_j(2k) \int_{\lfloor m \rfloor}^r \frac{v}{m^2} 2^{\frac{v(u-1)}{m^2}} du$$

Thus the terms in the summation are decreasing and

$$C_2(k, r) > 2 \sum_{j=r}^{m-1} h_j(2k) \int_{\lfloor m \rfloor}^r \frac{v}{m^2} 2^{\frac{v(u-1)}{m^2}} du$$

Note that this is increasing with $m$ and therefore $k$. Hence

$$C_2(k, r) > 2 \sum_{j=r}^{m-1} h_j(2k) \int_{\lfloor m \rfloor}^r \frac{v}{m^2} 2^{\frac{v(u-1)}{m^2}} du$$

An alternative lower bound for $C(k, r)$ can be found using Claim A.3. By this claim

$$C(k, r) = 1 - \sum_{j=r}^{2k} h_j(2k).$$

Let

$$C_3(k, r) := 1 - \sum_{j=r}^{2k} h_j(2k),$$

and assume that $r \geq \lceil m \rceil + 1$.

$$1 - C_3(k, r) = \sum_{j=r}^{2k} j \left[ \frac{2k}{2k} \right] \frac{\ln \left( 1 - \frac{j}{2k} \right)}{\ln(2)}$$

$$= \sum_{j=r}^{2k} j \exp \left( \sum_{t=1}^{j-1} \ln \left( 1 - \frac{t}{2k} \right) \right)$$

204
\[ \sum_{j=r}^{2k} j \exp \left( \sum_{t=1}^{j-1} \left( -\frac{t}{2k} \right) \right) = \sum_{j=r}^{2k} \frac{j}{2k} \exp \left( -\frac{j(j-1)}{4k} \right). \]

Let \( f(v) = ve^{-\frac{v(v-1)}{2m^2}} \). Then
\[ f'(v) = e^{-\frac{v(v-1)}{2m^2}} \left( 1 - \frac{2v^2 - v}{2m^2} \right), \]
and \( f'(v) < 0 \) if
\[ \frac{2v^2 - v}{2m^2} > 1 \]
\[ 2v^2 - v > 2m^2, \]
which holds for all \( v \geq m + 1 \) provided
\[ 2(m + 1)^2 - (m + 1) > 2m^2 \]
\[ 2m^2 + 4m + 2 - m - 1 > 2m^2, \]
which is true.

Hence, the terms in the summation are decreasing, and \( C_3(k, r) \) may be upper bounded by an integral from \( r - 1 \) to \( 2k = m^2 \):
\begin{align*}
1 - C_3(k, r) &< \int_{r-1}^{m^2} \frac{v}{m^2} e^{-\frac{v(v-1)}{2m^2}} dv \\
&= \int_{\rho}^{m} ue^{-\frac{u}{2m}} du \\
&< \int_{\rho}^{m} ue^{-\frac{u^2}{2} + \frac{m}{2}} du \\
&< e^{\frac{1}{2}} \int_{\rho}^{\infty} ue^{-\frac{u^2}{2}} du \\
&= e^{-\frac{\rho^2}{2} + 1}. \tag{A.15}
\end{align*}

Combining the bounds (A.12), (A.13), (A.14) and (A.15), let
\[ \tilde{C}_{1a}(m, \rho) := \frac{\rho^2}{2} - \frac{\rho^3}{6}, \]
\[ \tilde{C}_{1b}(m) := \frac{1}{2} \left( 1 - \frac{1}{m} \right)^2 - \frac{1}{6} \left( 1 - \frac{1}{m} \right)^2, \]

\[ \tilde{C}_2(m, \rho) := \tilde{C}_{1b}(m) + \frac{1}{\ln(2)} \left( \frac{m-1}{m} \right)^\frac{m-1}{2} \left( 2^{-\left(1+\frac{1}{m}\right)^2} - 2^{-\left(\rho+\frac{1}{m}\right)^2} \right), \]

\[ \tilde{C}_3(m, \rho) := 1 - e^{-\frac{\rho^2+1}{2}}, \]

and let

\[ \tilde{C}(m, \rho) := \begin{cases} 
\tilde{C}_{1a}(m, \rho) & \text{for } \rho \leq \frac{\lfloor m \rfloor - 1}{m} \\
\tilde{C}_{1b}(m) & \text{for } \rho = \frac{\lfloor m \rfloor}{m} \\
\max(\tilde{C}_2(m, \rho), \tilde{C}_3(m, \rho)) & \text{for } \rho \geq \frac{\lfloor m \rfloor + 1}{m}.
\end{cases} \quad (A.16) \]

Then for fixed \( \rho \), \( \tilde{C}(m, \rho) \) is increasing with \( m \) and for all \( k \geq 2 \) and \( r \in [k-1] \),

\[ C(k, r) > \tilde{C}(\sqrt{2k}, \frac{r - 1}{\sqrt{2k}}). \]

Combining (A.4), (A.11) and (A.16), let

\[ \tilde{N}(m, \rho) := \tilde{C}(m, \rho) + \frac{\tilde{A}(m) + \tilde{B}(m, \rho)}{m\rho + 3} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}. \quad (A.17) \]

Then for fixed \( \rho \), \( \tilde{N}(m, \rho) \) is increasing with \( m \) and for all \( k \geq 2 \) and \( r \in [k-1] \),

\[ N(k, r) > \tilde{N}(\sqrt{2k}, \frac{r - 1}{\sqrt{2k}}). \]

So, to show that \( N(k, r) \) is positive for all \( k \geq 18 \), it is sufficient to show that \( \tilde{N}(m, \rho) \) is positive for all \( m \geq 6 \) and \( \rho \in [0, \frac{m}{2}] \).

Define

\[ \tilde{N}_1(m, \rho) := \tilde{C}_{1a}(m, \rho) + \frac{\tilde{A}(m) + \tilde{B}_{1a}(m, \rho)}{m\rho + 3} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}, \]

\[ \tilde{N}_2(m, \rho) := \tilde{C}_{1a}(m, \rho) + \frac{\tilde{A}(m) + \tilde{B}_{1b}(m, \rho)}{m\rho + 3} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}, \]

\[ \tilde{N}_3(m, \rho) := \tilde{C}_{1b}(m) + \frac{\tilde{A}(m) + \tilde{B}_2(m, \rho)}{m\rho + 3} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}, \]

\[ \tilde{N}_4(m, \rho) := \tilde{C}_2(m, \rho) + \frac{\tilde{A}(m) + \tilde{B}_2(m, \rho)}{m\rho + 3} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}, \]

\[ \tilde{N}_5(m, \rho) := \tilde{C}_3(m, \rho) + \frac{\tilde{A}(m) + \tilde{B}_2(m, \rho)}{m\rho + 3} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}. \]
Then for fixed $\rho$, each of these is increasing with $m$. So, for any $m \geq 6$, with $m = \sqrt{2k}$ for integer $k \geq 18$,

$$\tilde{N}(m, \rho) \geq \begin{cases} 
\tilde{N}_1(6, \rho) & \text{for } 0 \leq \rho \leq \frac{[m]-2}{m}, \\
\tilde{N}_2(6, \rho) & \text{for } \rho = \frac{[m]-1}{m}, \\
\tilde{N}_3(6, \rho) & \text{for } \rho = \frac{[m]}{m}, \\
\max(\tilde{N}_4(6, \rho), \tilde{N}_5(6, \rho)) & \text{for } \frac{[m]+1}{m} \leq \rho \leq 3, \\
\tilde{N}_5(m', \rho) & \text{for } m' \leq m \text{ and } 3 < \rho \leq \frac{m}{2}.
\end{cases}$$

Now, for all $m \geq 6$,

$$\left[0, \frac{[m]-2}{m}\right], \left[\frac{[m]-1}{m}, \frac{m}{6}\right], \left[\frac{[m]}{m}, \frac{7}{6}\right], \left[\frac{[m]+1}{m}, 3\right] \subset [0, 1].$$

So, to show that $\tilde{N}(m, \rho)$ is positive on $[0, 3]$ for any $m \geq 6$, it is sufficient to show that each $\tilde{N}_i(6, \rho)$ is positive on the appropriate range above.

Firstly, for $\tilde{N}_1(6, \rho)$, note that $\tilde{B}_{1a}(m, \rho)$ and $\tilde{C}_{1a}(m, \rho)$ are increasing with $\rho$ for $0 < \rho < 2$, so for any interval $[\alpha, \beta] \subseteq [0, 1],$

$$\tilde{N}_1(6, \rho) \geq \tilde{C}_{1a}(6, \alpha) + \frac{\tilde{A}(6) + \tilde{B}_{1a}(6, \alpha)}{6\beta + 3} - \frac{\tilde{A}(6)}{\tilde{A}(6) - 2} \quad \text{for } \rho \in [\alpha, \beta]$$

$$=: \tilde{M}_1(\alpha, \beta).$$

Direct evaluation reveals

$$\tilde{M}_1(0, 0.4) > 0.0302,$$
$$\tilde{M}_1(0.4, 0.7) > 0.0514,$$
$$\tilde{M}_1(0.7, 1) > 0.0460.$$

Hence $\tilde{N}_1(6, \rho)$ is positive for all $\rho \in [0, 1]$.
Similarly, for \( \tilde{N}_2(6, \rho) \), and for any \([\alpha, \beta] \subseteq [\frac{5}{6}, 1]\),

\[
\tilde{N}_2(6, \rho) \geq \tilde{C}_{1a}(6, \alpha) + \frac{\tilde{A}(6) + \tilde{B}_{1b}(6)}{6\beta + 3} - \frac{\tilde{A}(6)}{\tilde{A}(6) - 2} \quad \text{for } \rho \in [\alpha, \beta]
\]

\[
=: \tilde{M}_2(\alpha, \beta).
\]

Direct evaluation reveals

\[
\tilde{M}_2\left(\frac{5}{6}, 1\right) > 0.1175,
\]

so \( \tilde{N}_2(6, \rho) \) is positive for all \( \rho \in [\frac{5}{6}, 1] \).

Next, for \( \tilde{N}_3(6, \rho) \) note that for any \([\alpha, \beta] \subset [1, 3]\),

\[
\tilde{B}_2(6, \rho) > \tilde{B}_{1b}(6) + 6 \frac{\frac{2}{\beta \ln(2)}}{6} \left(\frac{5}{6}\right)^{\frac{5}{6}} \left(2^{\beta(1+\frac{1}{6})} - 2^{\alpha(\alpha+\frac{1}{6})}\right) \quad \text{for } \rho \in [\alpha, \beta]
\]

\[
=: \tilde{B}_*(\alpha, \beta),
\]

so

\[
\tilde{N}_3(6, \rho) \geq \tilde{C}_{1b}(6) + \frac{\tilde{A}(6) + \tilde{B}_*(\alpha, \beta)}{6\beta + 3} - \frac{\tilde{A}(6)}{\tilde{A}(6) - 2} \quad \text{for } \rho \in [\alpha, \beta]
\]

\[
=: \tilde{M}_3(\alpha, \beta).
\]

Direct evaluation reveals

\[
\tilde{M}_3(1, 1.1) > 0.0086,
\]

\[
\tilde{M}_3(1.1, 1.167) > 0.0014.
\]

Hence \( \tilde{N}_3(6, \rho) \) is positive for all \( \rho \in [1, \frac{7}{6}] \).

Similarly, for \( \tilde{N}_4(6, \rho) \) and \( \tilde{N}_5(6, \rho) \), \( \tilde{C}_2(m, \rho) \) and \( \tilde{C}_3(m, \rho) \) are increasing with \( \rho \) so, for any \([\alpha, \beta] \subset [1, 3]\),

\[
\tilde{N}_4(6, \rho) \geq \tilde{C}_2(6, \alpha) + \frac{\tilde{A}(6) + \tilde{B}_*(\alpha, \beta)}{6\beta + 3} - \frac{\tilde{A}(6)}{\tilde{A}(6) - 2} \quad \text{for } \rho \in [\alpha, \beta]
\]

\[
=: \tilde{M}_4(\alpha, \beta),
\]

\[
\tilde{N}_5(6, \rho) \geq \tilde{C}_3(6, \alpha) + \frac{\tilde{A}(6) + \tilde{B}_*(\alpha, \beta)}{6\beta + 3} - \frac{\tilde{A}(6)}{\tilde{A}(6) - 2} \quad \text{for } \rho \in [\alpha, \beta]
\]

\[
=: \tilde{M}_5(\alpha, \beta).
\]
Direct evaluation reveals

\[ \tilde{M}_4(1, 1.1) > 0.0086, \]
\[ \tilde{M}_4(1.1, 1.2) > 0.0267, \]
\[ \tilde{M}_4(1.2, 1.3) > 0.0380, \]
\[ \tilde{M}_4(1.3, 1.4) > 0.0429, \]
\[ \tilde{M}_4(1.4, 1.5) > 0.0421, \]
\[ \tilde{M}_4(1.5, 1.65) > 0.0083, \]

and

\[ \tilde{M}_5(1.65, 1.9) > 0.0153, \]
\[ \tilde{M}_5(1.9, 2.4) > 0.0204, \]
\[ \tilde{M}_5(2.4, 3) > 0.0834. \]

Hence \( \max(\tilde{N}_4(6, \rho), \tilde{N}_5(6, \rho)) \) is positive for all \( \rho \in [1, 3] \).

Finally, it remains to be shown that \( \tilde{N}(m, \rho) \) is positive on \([3, m^2/2] \). Note that

\[ \frac{\sqrt{2}i}{2} + \frac{1}{\sqrt{2}i} > \frac{\sqrt{2}(i + 1)}{2}, \]

so by induction

\[ \left[3, \frac{m^2}{2}\right] \subset \bigcup_{i=18}^{m^2/2} \left[\frac{\sqrt{2}i}{2}, \frac{\sqrt{2}i}{2} + \frac{1}{\sqrt{2}i}\right]. \]

Let

\[ \tilde{N}_6(m) := \tilde{C}_3(m, \frac{m}{2}) + \tilde{A}(m) + \tilde{B}_{1k}(m) \]
\[ \quad + \frac{\tilde{A}(m) + 3}{m(\frac{m}{2} + \frac{1}{m})} - \frac{\tilde{A}(m)}{\tilde{A}(m) - 2}. \]

Then

\[ \tilde{N}_5(m, \rho) > \tilde{N}_6(m) \quad \text{for} \quad \rho \in \left[\frac{m}{2}, \frac{m}{2} + \frac{1}{m}\right], \]

and, for each integer \( i \geq 18 \) such that \( \sqrt{2}i < m \),

\[ \tilde{N}(m, \rho) \geq \tilde{N}_5(m, \rho) \geq \tilde{N}_5(\sqrt{2}i, \rho) > \tilde{N}_6(\sqrt{2}i) \quad \text{for} \quad \rho \in \left[\frac{\sqrt{2}i}{2}, \frac{\sqrt{2}i}{2} + \frac{1}{\sqrt{2}i}\right], \]

209
since $\tilde{N}_5(m, \rho)$ is increasing with $m$ for fixed $\rho \geq \lceil \frac{m+1}{m} \rceil$. Hence, to show that $\tilde{N}(m, \rho)$ is positive on $[3, \frac{m}{2}]$ it is sufficient to show that $\tilde{N}_6(m')$ is positive for all $6 \leq m' < m$.

Now, $\tilde{N}_6(m)$ is positive if and only if
\[
e^{-\frac{m^2}{\pi} + \frac{1}{2}} < \frac{m \left( \frac{3}{4} - \frac{1}{m^2} + \frac{1}{m^2} \right) + \sqrt{\frac{2}{\pi}} m}{m \left( \frac{\sqrt{2}}{2} + \frac{1}{m} \right) + 3} + 1 - \frac{\sqrt{\frac{2}{\pi}} m}{\sqrt{\frac{\pi}{2}} m - 2}
= \frac{\frac{3}{4} m - 1 + \frac{1}{m} + \sqrt{\frac{2}{\pi}} m}{m^2 + 4} - \frac{2}{\sqrt{\frac{\pi}{2}} m - 2},
\]
which is equivalent to
\[
\left( \frac{m^2}{2} + 4 \right) e^{-\frac{m^2}{\pi} + \frac{1}{2}} < \left( \frac{3}{4} + \sqrt{\frac{2}{\pi}} \right) m - 1 + \frac{1}{m} - \frac{m^2 + 8}{\sqrt{\frac{\pi}{2}} m - 2}
= \left( \frac{3}{4} + \sqrt{\frac{2}{\pi}} \right) m - 1 - \sqrt{\pi} m - 4 \frac{4}{\pi} - \frac{2 + (8 + \frac{8}{\pi} - \sqrt{\pi}) m}{\sqrt{\frac{\pi}{2}} m^2 - 2m}
= \left( \frac{3}{4} + \sqrt{\frac{2}{\pi}} - \sqrt{\frac{2}{\pi}} \right) m - 1 - 4 \frac{4}{\pi} - \frac{2}{\sqrt{\frac{\pi}{2}} m^2 - 2m} - 8 + \frac{8}{\pi} - \sqrt{\pi}.
\]
The right hand side of this is increasing with $m$ for $m > \sqrt{\frac{2}{\pi}}$, and the left hand side is decreasing for $m > 0$. For $m = 6$,
\[e^{-\frac{m^2}{\pi} + \frac{1}{2}} = e^{-4} < 0.403\]
and
\[
\left( \frac{3}{4} + \sqrt{\frac{2}{\pi}} - \sqrt{\frac{2}{\pi}} \right) m - 1 - 4 \frac{4}{\pi} - \frac{2}{36 \sqrt{\frac{2}{\pi}} - 12} - \frac{8 + \frac{8}{\pi} - \sqrt{\pi}}{6 \sqrt{\frac{2}{\pi}} - 2} > 3.215.
\]
Hence, $\tilde{N}_6(6)$ is positive and $\tilde{N}_6(m)$ is positive for all $m \geq 6$ (in fact, $\tilde{N}_6(m)$ is positive for all $m \geq \sqrt{26}$ or $k \geq 13$).

Hence, $\tilde{N}(m, \rho)$ is positive for all $m \geq 6$ and $\rho \in [0, \frac{m}{2}]$, so $N(k, r)$ is positive for all $k \geq 18$ and $r \in [k - 1]$. This completes the proof of Lemma 6.27 for $k \geq 18$.

Tables A.3 and A.3 give the values of $N(k, r)$ for $k = 7, \ldots, 17$ and each $r \in [k - 1]$. All values are positive, which completes the proof of Lemma 6.27.
Table A.1: $N(k, r)$ evaluated for $k = 7, \ldots, 17$ and $r = 1, \ldots, 8$

<table>
<thead>
<tr>
<th>$k$, $r$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0.3625</td>
<td>0.1617</td>
<td>0.1044</td>
<td>0.1200</td>
<td>0.1624</td>
<td>0.1998</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.5322</td>
<td>0.2979</td>
<td>0.2135</td>
<td>0.2073</td>
<td>0.2359</td>
<td>0.2690</td>
<td>0.2878</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.6817</td>
<td>0.4170</td>
<td>0.3088</td>
<td>0.2832</td>
<td>0.2983</td>
<td>0.3253</td>
<td>0.3452</td>
<td>0.3479</td>
</tr>
<tr>
<td>10</td>
<td>0.8164</td>
<td>0.5237</td>
<td>0.3940</td>
<td>0.3511</td>
<td>0.3534</td>
<td>0.3734</td>
<td>0.3921</td>
<td>0.3986</td>
</tr>
<tr>
<td>11</td>
<td>0.9397</td>
<td>0.6208</td>
<td>0.4716</td>
<td>0.4129</td>
<td>0.4033</td>
<td>0.4158</td>
<td>0.4320</td>
<td>0.4402</td>
</tr>
<tr>
<td>12</td>
<td>1.0538</td>
<td>0.7104</td>
<td>0.5432</td>
<td>0.4700</td>
<td>0.4492</td>
<td>0.4542</td>
<td>0.4670</td>
<td>0.4756</td>
</tr>
<tr>
<td>13</td>
<td>1.1604</td>
<td>0.7938</td>
<td>0.6099</td>
<td>0.5234</td>
<td>0.4920</td>
<td>0.4896</td>
<td>0.4984</td>
<td>0.5065</td>
</tr>
<tr>
<td>14</td>
<td>1.2609</td>
<td>0.8722</td>
<td>0.6726</td>
<td>0.5736</td>
<td>0.5323</td>
<td>0.5227</td>
<td>0.5272</td>
<td>0.5339</td>
</tr>
<tr>
<td>15</td>
<td>1.3561</td>
<td>0.9462</td>
<td>0.7319</td>
<td>0.6212</td>
<td>0.5705</td>
<td>0.5539</td>
<td>0.5540</td>
<td>0.5588</td>
</tr>
<tr>
<td>16</td>
<td>1.4467</td>
<td>1.0166</td>
<td>0.7882</td>
<td>0.6666</td>
<td>0.6070</td>
<td>0.5837</td>
<td>0.5793</td>
<td>0.5818</td>
</tr>
<tr>
<td>17</td>
<td>1.5334</td>
<td>1.0837</td>
<td>0.8420</td>
<td>0.7099</td>
<td>0.6420</td>
<td>0.6122</td>
<td>0.6032</td>
<td>0.6032</td>
</tr>
</tbody>
</table>

Table A.2: $N(k, r)$ evaluated for $k = 10, \ldots, 17$ and $r = 9, \ldots, 16$

<table>
<thead>
<tr>
<th>$k$, $r$</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.3886</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.4350</td>
<td>0.4154</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.4736</td>
<td>0.4588</td>
<td>0.4324</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.5065</td>
<td>0.4954</td>
<td>0.4734</td>
<td>0.4425</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.5351</td>
<td>0.5269</td>
<td>0.5085</td>
<td>0.4812</td>
<td>0.4475</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.5604</td>
<td>0.5543</td>
<td>0.5390</td>
<td>0.5150</td>
<td>0.4842</td>
<td>0.4490</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.5832</td>
<td>0.5785</td>
<td>0.5657</td>
<td>0.5447</td>
<td>0.5167</td>
<td>0.4838</td>
<td>0.4481</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.6039</td>
<td>0.6002</td>
<td>0.5895</td>
<td>0.5710</td>
<td>0.5456</td>
<td>0.5150</td>
<td>0.4809</td>
<td>0.4454</td>
</tr>
</tbody>
</table>
# Appendix B

## Index of symbols and notation

### General symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{R}$</td>
<td>real numbers</td>
<td>18</td>
</tr>
<tr>
<td>$\mathbb{Z}$</td>
<td>integers</td>
<td>99</td>
</tr>
<tr>
<td>$\mathbb{N}$</td>
<td>non-negative integers</td>
<td>20</td>
</tr>
<tr>
<td>$[x]$</td>
<td>${1, 2, \ldots, x}$, for integer $x$</td>
<td>14</td>
</tr>
<tr>
<td>$[\ldots]$</td>
<td>the event described by “…”</td>
<td>16</td>
</tr>
<tr>
<td>$[r]$</td>
<td>integer part of $r$ (in Lemma 3.7 only)</td>
<td>40</td>
</tr>
<tr>
<td>$\lfloor r \rfloor$</td>
<td>floor of $r$</td>
<td>53</td>
</tr>
<tr>
<td>$\lceil r \rceil$</td>
<td>ceiling of $r$</td>
<td>8</td>
</tr>
<tr>
<td>$I_A$</td>
<td>indicator variable for event $A$</td>
<td>16</td>
</tr>
<tr>
<td>$\in$</td>
<td>element of OR has/with the distribution of</td>
<td>15</td>
</tr>
<tr>
<td>$\land$</td>
<td>and</td>
<td>39</td>
</tr>
<tr>
<td>$\lor$</td>
<td>or</td>
<td>64</td>
</tr>
<tr>
<td>$\neg$</td>
<td>not</td>
<td>45</td>
</tr>
<tr>
<td>$P$</td>
<td>probability</td>
<td>16</td>
</tr>
<tr>
<td>$E$</td>
<td>expectation</td>
<td>16</td>
</tr>
<tr>
<td>$P(A \mid B)$</td>
<td>probability of $A$ conditional on $B$</td>
<td>16</td>
</tr>
<tr>
<td>u.a.r.</td>
<td>uniformly at random</td>
<td>16</td>
</tr>
<tr>
<td>a.a.s.</td>
<td>asymptotically almost surely</td>
<td>17</td>
</tr>
</tbody>
</table>
$O()$ big-O Landau symbol ........................................ 17
do() small-o Landau symbol ........................................ 17
\Theta() $f = \Theta(g)$ if $f = O(g)$ and $g = O(f)$ ............... 17
\sim asymptotic to ........................................ 17
$\delta_{i,j}$ the Kronecker delta ........................................ 88

Chapter 1

$\mathcal{G}(n, m)$ the uniform random graph ..................... 14
$\mathcal{G}(n, p)$ the binomial random graph ....................... 14
$\mathcal{H}_t$ history of a given process up to time $t$ .......... 18

Chapter 2

Core$(k)$ an algorithm to find the $k$-core of a pseudograph. See Alg’m 2.1 22
Load$(k)$ a load balancing algorithm for load $k$. See Alg’m 2.2 .......... 22

Chapter 3

$\mathcal{A}(n, k)$ the random allocation. See Def’n 3.1 ............ 30
$\mathcal{M}(n, m)$ the random pseudograph. See Def’n 3.1 ............ 30
$D_s^d$ := \{(h_1, \ldots, h_d) : \sum_{i=1}^d h_i = s\} ..................... 31
Z(\lambda) Poisson variable with mean \lambda .......................... 31
L_i^d := \{(h_1, \ldots, h_d) : 0 \leq h_i < l, i = 1 \ldots, d\} .......... 33
D_d := [ d(i) = d_i, 1 \leq i \leq r \text{ and } d(i) \geq l, i \geq r + 1 ] .......... 33
$\mathcal{M}_l(d, \nu, s)$ the conditioned pseudograph. See Def’n 3.2 ........ 33
$Z^{(l)}(\lambda)$ truncated Poisson variable. See (3.4) .................. 34
$f_a(\lambda) := \sum_{i \geq a} \lambda^i / i! \ldots................. 34
\psi_l(\lambda) := \lambda f_{l-1}(\lambda) / f_l(\lambda) \ldots................. 34
H_{l,s}^d := \{(h_1, \ldots, h_d) : \sum h_i = s \text{ and } h_i \geq l\} ............. 35
\( \text{Multi}(l, d, s) \) restricted multinomial distribution. See Def’n 3.3 .......................... 35
\( \lambda_{l,c} \) the positive root of the equation \( \psi_l(\lambda) = c \) ................................. 38
\( p_{l,c,j} := \lambda_{l,c}^j/(f_l(\lambda_{l,c}) j!) \) ................................................. 38
\( m_{l,d,s,j} := \mathbf{P}(h_1 = j \mid (h_1, \ldots, h_d) \in \text{Multi}(l, d, s)) \) ......................................... 44

Chapter 4

\( \Omega_{\text{Core}(k)} \) the process determined by \( \text{Core}(l) \) ........................................... 47
\( \Omega_{\text{Load}(k)} \) the process determined by \( \text{Load}(l) \) ........................................... 47
\( \text{ACore}(k) \) an algorithm that models \( \text{Core}(k) \). See Alg’m 4.21 ..................... 48
\( \text{ALoad}(k) \) an algorithm that models \( \text{Load}(k) \). See Alg’m 4.24 ..................... 48
\( \Omega_{\text{ACore}(k)} \) the process determined by \( \text{ACore}(l) \) ........................................... 49
\( \Omega_{\text{ALoad}(k)} \) the process determined by \( \text{ALoad}(l) \) ........................................... 49
d(\( v \)) degree of \( v \) ................................................................. 50
d^L(\( v \)) load degree of \( v \) .......................................................... 50
d^−(\( v \)) in-degree of \( v \) ........................................................... 50
\( \text{Pri}_k \) priority function for vertices. See Def’n 4.1 .................................................. 50
\( \text{EdgeDel}(k) \) an edge deletion step ................................................................. 50
\( \text{EdgeOri}(k) \) an edge orientation step ............................................................. 51
\( \hat{L}_l^r \) .................................................................
\( \mathcal{D}_{d,\mu,s} := [d(i) = d_i, 1 \leq i \leq r \land d(i) \geq l, i \leq r + 1 \land \sum_{i=r+1}^{\mu} d(i) = s] \) .... 54
\( \hat{D}_{d,j,\mu,s} \) .................................................................
\( \text{Unor}(M) \) the unoriented part of \( M \). See Def’n 4.4 ................................. 54
\( \Omega_c \) the process determined by \( \text{EdgeDel}(l) \) ........................................... 57
\( p_{\mathcal{M}_t}(t) := \mathbf{P}(M_t \in \mathcal{M}_t \mid M_0 \in \mathcal{M}_0) \) ........................................... 57
\( \text{T}_{\text{Ed}(k)} \) transition probability w.r.t. \( \text{EdgeDel}(k) \). See Def’n 4.6 .......................... 57
\( \text{G}_c \) the graph of the \( \Omega_c \). See Def’n 4.7 ..................................................... 58
\( \mathbf{e}^{(j)} \) vector with \( j \)th term 1 and zeros elsewhere .................................................. 59
\( \tilde{\mathcal{M}}_t(\mathbf{d}, j, \nu, s) \) the conditioned, partially oriented pseudograph. See Def’n 4.9 63
\( \tilde{E}_l := \bigcup_{r \geq 1} \{(\mathbf{d}, j) \in \hat{L}_l^r : d_i - 2j_i = 0 \lor d_i = l, \; i = 1, \ldots, r\} \) ............. 64
\[ \Omega_L \]
the process determined by \textbf{EdgeOri}(l) .......................... 64
\[ G_L \]
the graph of the \( \Omega_L \) .................................................. 64
\[ T_{\text{EO}(k)} \]
transition probability w.r.t. \textbf{EdgeOri}(k). See Def’n 4.10 ...... 64
\[ \text{GenDel}(k) \]
an edge generation and deletion step .............................. 72
\[ \text{GenOri}(k) \]
an edge generation and orientation step ............................ 73
\[ A(d) \]
pre-allocation. See Def’n 4.12 ........................................ 69
\[ A(d,j) \]
partially oriented pre-allocation. See Def’n 4.13 ................. 71
\[ B(d, \nu, s) \]
partial pre-allocation. See Def’n 4.14 ............................... 71
\[ \hat{B}(d,j, \nu, s) \]
partially oriented partial pre-allocation. See Def’n 4.15 .......... 72
\[ q_{\nu,s,l} := l\nu m_{\nu,s,l}/s \] ............................ 72
\[ T_{\text{GD}(k)} \]
transition probability w.r.t. \textbf{GenDel}(k). See Def’n 4.16 ...... 73
\[ T_{\text{GO}(k)} \]
transition probability w.r.t. \textbf{GenOri}(k). See Def’n 4.18 ...... 75
\[ T_s \]
time when the process becomes constant. See (4.21) and (4.24) . 76
\[ G_{\text{AC}} \]
the graph of the \( \Omega_{\text{ACore}(k)} \) .................................. 78
\[ T_f \]
: = m .................................................. 78
\[ L_{l_1,l_2}^d \]
: = \{ (h_1, \ldots, h_d) : l_1 \leq h_i < l_2, i = 1, \ldots, d \} ........... 80

\chapter{Chapter 5}

\[ h_k(\mu) \]
: = \( \mu e^{\mu} / f_{k-1}(\mu) \) ........................................ 83
\[ c_k \]
: = \inf\{ h_k(\mu) : \mu > 0 \} .................................... 83
\[ \mu_{k,c} \]
: = \max\{ \mu : h_k(\mu) = c \}, when it exists ..................... 83
\[ d^{(0)} \]
\( \in \mathbb{L}_k^r \). Low degree sequence of \( M_0 \in \mathcal{M}(n, m) \) ............ 84
\[ s_0 \]
: = \( 2m - \sum_i d_i^{(0)} \) ......................................... 84
\[ B(0) \]
: = \( B(d^{(0)}, n - r_0, s_0) \) ................................... 84
\[ c \]
: = \( 2m/n \) .................................................. 84
\[ \{ B(t) \}_{t \geq 0} \]
process determined by \( B(0) \) and \textbf{ACore}(k) ................. 84
\[ T_s \]
: = \min\{ T \geq 0 : B(t) = B(T) for all t \geq T \} ............... 84
\[ L(t) \]
number of free points in \( B(t) \) ................................ 84
\[ Y_i(t) \]
r.v. for number of vertices of degree \( i \) in \( B(t) \) ................. 84

215
Chapter 6

The solution to \( \psi_k(\lambda) = 2k \) .................................................. 96

\( \rho_k \) := \( h_{k+1}(\lambda_k) \) .......................................................... 96

\( M_0 \) \((k+1)\)-core of \( M \in \mathcal{M}(n_0, m_0) \) ........................................ 98

\( \rho \) := \( 2m/n \), the density of \( M_0 \) ............................................. 98

\( \hat{B}(0) \) partial pre-allocation corresponding to \( M_0 \) ................. 98

\( \{\hat{B}(t)\}_{t \geq 0} \) process determined by \( \hat{B}(0) \) and \textbf{ALoad}(k) ......................... 98

\( T_s \) := \( \min\{T \geq 0 : \hat{B}(t) = \hat{B}(T) \text{ for all } t \geq T\} \) ......................... 98

\( D \) := \{\( (d,j) \in \mathbb{Z}^2 : 0 \leq d \leq 2k + 1, 0 \leq j \leq \lfloor \frac{d}{2} \rfloor \} \setminus \{2k + 1, 0\} \) ............... 99

\( Y_{d,j}(t) \) number of vertices, \( i \), in \( \hat{B}(t) \) with \( d^L(i) = d \) and \( d^-(i) = j \) .... 99

\( Y \) vector of the \( Y_a \), for fixed ordering of the \( a \) ............................. 99

\( W(t) \) r.v. for \( n - r(t) \) in \( \hat{B}(t) = \hat{B}(d(t), j(t), n - r(t), s(t)) \) ...................... 99

\( H(t) \) r.v. for \( s(t) \), the number of points in vertex \( n + 1 \) of \( \hat{B}(t) \) ................. 99

\( L(t) \) number of free points in \( \hat{B}(t) \) .............................................. 99

\( A \) := \( \{(d,j) \in D : d < 2k \text{ and } d - j \leq k\} \) .................................. 100

\( B \) := \( \{(d,j) \in D : k < d < 2k \text{ and } d - j > k\} \) .................................. 100

\( C \) := \( \{(d,j) \in D : d = 2k + 1\} \) ............................................ 100

\( X_{l}(t) \) := \( \sum_{(d,j) \in A} (d - 2j)Y_{d,j}(t) \) ........................................... 100

\( X(t) \) := \( \sum_{(d,j) \in B} (2k - d)Y_{d,j}(t) + X_{l}(t) \) ........................................... 100

\( Z(t) \) := \( H(t) - 2kW(t) + \sum_{(d,j) \in C} Y_{d,j} \) ........................................... 100

\( T_h \) := \( \min\{t : X(t) = 0\} \) ...................................................... 100

\( T_l(\epsilon_b) \) := \( \min\{t : \phi_l(Y(t)) > 1 - \epsilon_b\} \). See Def’n 6.13 ......................... 102

\( T_g(\epsilon_a) \) := \( \min\{t : g(X, Z, H) > \epsilon_a\} \). See Def’n 6.16 ......................... 103
Chapter 7

$C_1(t) := [d_2 \leq 2k \text{ and } d_2 - j_2 > k \text{ at time } t]$ .................................................. 105
$C_2(t) := [d_2 = 2k + 1 \text{ or } i_2 = n + 1 \text{ at time } t]$ ........................................ 105
$C_3(t) := [d_2 - j_2 \leq k \text{ at time } t]$ ...................................................... 105
$\phi_l(Y) := \frac{1}{2} \sum_{r=1}^k r(r+1)Y_{2k-r+1,k-r}$. See Def’n 6.11 .................. 108
$g(x, z, h) := \frac{x}{z}(1 - \frac{2z}{k}) - 1$ .................................................. 116
$X_w := \sum_{j=0}^{w-2} Y_{k+w-1,j}$ .................................................. 125
$T_w := \min \{t: \sum_{u=1}^w X_u(t) = 0\}$ .................................................. 125
$\phi_1(Y)$ .................................................. 125
$\phi_w(Y) := \phi_l(Y) + \frac{1}{2} \sum_{j=0}^{w-2} (k + w - 2j)Y_{k+w,j}$ .................................. 125
$\phi(Y(t)) := \sum_{w=1}^{k-1} \phi_w(Y(t))I_{T_w \leq t < T_{w+1}}$ .................................. 125
$T_e(\epsilon_b) := \min \{t: \phi(Y) > 1 - \epsilon_b\}$ .................................. 125
\( \text{Phase } w := \{t: t < T_e \text{ and } T_w \leq t < T_{w+1}\} \text{ for } 1 \leq w < k \) ......... 126
\( \text{Phase } k := \{t: t < T_e \text{ and } T_k \leq t \leq T_f\} \) ........................................ 126
\( \text{Explosion phase} = \{t: T_e \leq t \leq T_f\} \) ........................................ 126

\[ M_F \text{ a matching of points derived from } F_t. \text{ See Def’n 7.4 .............. 145} \]
\[ F_t \text{ a graph on points in an allocation, induced by } A\text{Load}(k) ........ 145 \]
\[ X_T := X(T), \text{ number of blue points} ........................................... 149 \]
\[ Z_T := Z(T) .................................................. 149 \]
\[ H_T := H(T), \text{ number of red points} ....................................... 149 \]
\[ W_T := W(T) .................................................. 149 \]
\[ \hat{A}_T \text{ a partially oriented pre-allocation obtained from } \hat{B}(T) ......... 149 \]
\[ d_T(i) \text{ load degree of vertex } i \text{ in } \hat{A}_T ......................................... 151 \]
\[ j_T(i) \text{ in-degree of vertex } i \text{ in } \hat{A}_T ........................................... 151 \]
\[ \delta(i) := \min(2k - d_T(i), d_T(i) - 2j_T(i)) ....................................... 151 \]
\[ P_R \text{ a partial pairing derived from } M_F ........................................ 159 \]
\[ G_R \text{ a graph derived from an allocation and the induced } P_R ........ 159 \]
\[ U_f(A) \text{ Event on the degree sequence of } A. \text{ See Def’n 7.9 ............. 160} \]

217
## Chapter 9

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>EST</td>
<td>Euclidean Steiner tree</td>
</tr>
<tr>
<td>MST</td>
<td>minimum spanning tree</td>
</tr>
<tr>
<td>MStT</td>
<td>minimal Steiner tree</td>
</tr>
<tr>
<td>StT</td>
<td>Steiner tree</td>
</tr>
<tr>
<td>MA</td>
<td>metropolis algorithm</td>
</tr>
<tr>
<td>StT(P)</td>
<td>Steiner tree on terminal set $P$</td>
</tr>
<tr>
<td>KGT</td>
<td>Kruskal greedy tree</td>
</tr>
<tr>
<td>PGT</td>
<td>Prim greedy tree</td>
</tr>
<tr>
<td>JoinOp</td>
<td>joining operation</td>
</tr>
<tr>
<td>$C(f)$</td>
<td>cost of JoinOp $f$</td>
</tr>
<tr>
<td>$L(T)$</td>
<td>the sum of the lengths of all the edges in tree $T$</td>
</tr>
<tr>
<td>ES JoinOp</td>
<td>edge short joining operation</td>
</tr>
<tr>
<td>KGS($P$)</td>
<td>Kruskal Greedy Steiner tree on terminal set $P$</td>
</tr>
<tr>
<td>PGS1($P$)</td>
<td>Prim Greedy Steiner tree on $P$, version 1</td>
</tr>
<tr>
<td>PGS2($P$)</td>
<td>Prim Greedy Steiner tree on $P$, version 2</td>
</tr>
<tr>
<td>KGTopt</td>
<td>Kruskal greedy tree with repositioning as final step</td>
</tr>
<tr>
<td>PGTTopt</td>
<td>Prim greedy tree with repositioning as final step</td>
</tr>
</tbody>
</table>
Appendix C

Index of terminology

allocate, an edge .......................................................... 10
allocation, allocation model .............................................. 30
asymptotically almost surely ......................................... 17
asymptotically almost surely optimal ............................... 13
branching process ......................................................... 20
cost, of a joining operation .............................................. 178
d-dimensional Euclidean Steiner tree problem ....................... 172
degree, of a vertex .......................................................... 50
density, of a pseudograph ................................................ 14
edge short joining operation ............................................. 179
Euclidean Steiner tree problem ........................................... 172
event ................................................................. 16
explosion ................................................................. 110
explosion phase ........................................................... 126
exponentially small tail ................................................... 21
expose, a point or the mate of a point ................................ 70
final state .............................................................. 16
free point ............................................................. 50
full topology .............................................................. 175
Galton-Watson branching process ........................................... 20
graph .............................................................................. 14
heavy vertices ................................................................. 52
history, of a random process .............................................. 18
in-degree, of a vertex ....................................................... 50
in-point ............................................................................. 50
joining operation ............................................................... 178
junction ............................................................................. 178
$k$-core .............................................................................. 9
$K, c$ tail ........................................................................... 21
Kruskal Greedy Steiner tree ............................................... 181
length, of a tree ............................................................... 179
Lipschitz condition ............................................................ 18
load ................................................................................... 7
load, on a vertex ............................................................... 10
load degree, of a vertex ..................................................... 50
low degree sequence .......................................................... 53
low load-degree sequence .................................................. 53
Markov chain ................................................................. 23
mate, of a point .................................................................. 49
metropolis algorithm ......................................................... 173
minimal Steiner tree .......................................................... 172
minimum spanning tree ..................................................... 172
multigraph ......................................................................... 14
out-point ............................................................................. 50
pairing ................................................................................. 51
partially oriented pre-allocation ......................................... 71
partially oriented partial pre-allocation ............................... 72
partially oriented pseudograph .......................................... 50
partial pre-allocation ................................................. 71
phase $w$ ................................................................. 126
pre-allocation ......................................................... 69
priority, of a vertex .................................................. 50
priority 1 vertex ....................................................... 26
process, a vertex ....................................................... 76
process, a point ....................................................... 76
pseudograph ............................................................ 14
random graph .......................................................... 14
random matching ....................................................... 146
random process, discrete time ...................................... 18
random process, pseudograph or graph ......................... 15
random variable ....................................................... 16
represent, a red or blue point ..................................... 157
restriction, of a probability space ................................ 16
sharply concentrated threshold .................................... 9
simulated annealing .................................................. 183
size, of a pseudograph ............................................... 14
Steiner point ............................................................ 172
Steiner tree ............................................................. 173
step, of an algorithm or process ................................... 27
step $t$ ........................................................................ 76
stochastically dominates ............................................. 20
stopping time ............................................................ 18
successful output, of $\text{ACore}()$ or $\text{ALoad}()$ .................. 27
survival, of a point ..................................................... 150
terminal spaces ......................................................... 57
terminals, set of ......................................................... 172
threshold ................................................................. 9
transition probability .................................................. 57

topolgy ................................................................. 175

uniformly at random .................................................. 16

unoriented part, of a pseudograph .................................. 54

weight, of an edge in $G_c$ etc. ...................................... 58
Page i, 2nd paragraph:
Attributed pseudograph model to Bollobás and Frieze [7], as well as Chvátal [10]. Also on pages 2, 3 and 30.

Page 10, 2nd paragraph: Added reference to Cooper’s new result on the size of cores.

Page 12, before Lemma 1.1: Added reference to Hakimi [18] for this lemma.

Page 30, after Definition 3.1: Added explanation for why $\mathcal{M}(n, m)$ is not the uniform probability space of pseudographs and why this doesn’t matter. Added a new paragraph motivating the use of pseudographs instead of simple graphs.

Page 30, before the definition of $D_{d}^{s}$: Added new paragraph motivating the definition of $D_{d}^{s}$ and other similar sets of vectors.

Page 32, final paragraph in Section 3.1: Removed incorrect argument for the $X_{j}$ being sharply concentrated. Added new paragraph referencing the sharp concentration result in [9] and foreshadowing the use of random variables similar to $X_{j}$ in ensuing chapters.

Page 32, before the definition of $L_{d}^{j}$: Motivate definition for space of vectors $L_{d}^{j}$.

Page 33, 2nd and 3rd paragraphs: Added two new paragraphs explaining the meaning of “low degree sequence” and beginning the explanation for why we only consider pseudographs in which the low degree vertices are the first vertices in the degree sequence.

Page 34, after Definition 3.2: Explicitly state that the $\nu$ vertices of degree $l$ or greater in the definition of $\mathcal{M}_{l}(d, \nu, s)$ are the last $\nu$ vertices in the degree sequence. Added two new paragraphs continuing the explanation for why it’s sufficient to only consider these restricted spaces of pseudographs. One way of thinking of this is that a permutation of the labels on the vertices does not effect the distribution of the random pseudographs we are using.

Page 35, before the definition of $H_{l,s}^{d}$: Added a new paragraph motivating the definition of $H_{l,s}^{d}$. 
Page 49, 2nd paragraph of Section 4.1: Rewrote this paragraph relating the idea of permuting the labels on the points in vertices to that of permuting the labels on the vertices and how neither of these affects the distribution of the pseudographs involved.

Page 51, 1st paragraph of Section 4.2: Added a sentence referring back to discussion in previous section, which motivates this lemma.

Page 53, 3rd paragraph: Some rewording and motivation for the definition of $\hat{L}_t$.

Page 172, 3rd paragraph: Corrected statement about EST problem being NP-hard and put in the correct reference.

Page 173, 4th paragraph: Added two paragraphs commenting on how random terminal sets may differ from real world examples and referencing some know results for random terminal sets.
Author/s: 
Cain, Julie A

Title: 
Random graph processes and optimisation

Date: 
2006-01

Citation: 

Publication Status: 
Unpublished

Persistent Link: 
http://hdl.handle.net/11343/39113

File Description: 
Random graph processes and optimisation

Terms and Conditions: 
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.