

Efficient Counting of Degree Sequences

Kai Wang*

Department of Computer Sciences, Georgia Southern University, Statesboro, GA, USA, 30458

Abstract

Novel dynamic programming algorithms to count the set $D(n)$ of zero-free degree sequences of length n , the set $D_c(n)$ of degree sequences of connected graphs on n vertices and the set $D_b(n)$ of degree sequences of biconnected graphs on n vertices exactly are presented. They are all based on a recurrence of Barnes and Savage and shown to run in polynomial time and are asymptotically much faster than the previous best known algorithms for these problems. These appear to be the first polynomial time algorithms to compute $|D(n)|$, $|D_c(n)|$ and $|D_b(n)|$ to the author's knowledge and have enabled us to tabulate them up to $n = 118$, the majority of which were unknown. The available numerical results of $|D(n)|$ tend to give more supporting evidence of a conjecture of Gordon F. Royle about the limit of $|D(n)|/|D(n-1)|$. The OEIS entries that can be computed by algorithms in this paper are A004251, A007721, A007722 and A095268.

Keywords: dynamic programming, degree sequence, graphical partition, connected, biconnected, recurrence, counting, enumeration

1. Introduction

The method to count the number of unlabeled graphs, unlabeled connected graphs and unlabeled biconnected graphs on n vertices exactly based on the Redfield-Pólya enumeration theorem is well-known and proves to be efficient [22, 21, 23, 10]. However, there was no known efficient method to count the number of degree sequences of these graphs exactly. We present a solution to these problems in this paper. Only simple (i.e. with no loops or multiple edges) undirected graphs and their degree sequences are considered.

A degree sequence of length n is a sequence of nonnegative integers d_1, d_2, \dots, d_n (in which order is irrelevant so that they are usually written in non-increasing order without loss of generality) such that there is a simple undirected graph G of n vertices with the vertex degrees d_1, d_2, \dots, d_n . A zero-free degree sequence is a degree sequence such that each term of the sequence is positive, or equivalently, a degree sequence of a simple undirected graph with no isolated vertices. For convenience, from now on by degree sequences we mean *zero-free* degree sequences unless otherwise noted. Let $D(n)$ denote the set of degree sequences of length n . A positive integer sequence a_1, a_2, \dots, a_n (in which order is also irrelevant) with sum $N = \sum_{i=1}^n a_i$ is called a graphical partition of N with n parts, or a graphic sequence, if

*Corresponding author

Email address: kwang@georgiasouthern.edu (Kai Wang)

it is the degree sequence of a simple undirected graph. Note that N must be even to have a graphical partition, since the sum of the vertex degrees of a simple undirected graph is twice the number of its edges and therefore must be even. Sierksma and Hoogeveen [28] showed seven criteria to be equivalent to characterize graphic integer sequences, of which the Erdős-Gallai criterion [7] is probably the most widely known.

The problems of counting or enumerating the set $G(N)$ of graphical partitions of a fixed positive even integer N and the set $D(n)$ for a fixed length n have been extensively studied, for example [8, 27, 3, 4, 19, 16, 6, 14, 20]. Pittel [19] affirmatively resolved a problem originally posed in 1982 by Herbert S. Wilf whether the fraction of graphical partitions in the set $P(N)$ of all partitions of a positive integer N goes to 0 as N approaches infinity. Erdős and Richmond [8] showed that this fraction as a function of N cannot tend to 0 faster than $\pi/\sqrt{6N}$. Recently Pittel [20] has obtained an upper bound of $\exp(-0.11 \log N / \log \log N)$ for $|G(N)|/|P(N)|$ with N sufficiently large using a local limit theorem. Comparison between the bound and actual values of $|G(N)|/|P(N)|$ up to $N = 910$ suggests that the bound does not appear to be tight. Up till now, the asymptotic order of $|G(N)|$ is still unknown, though the order of $|P(N)|$ has been known to be $|P(N)| \sim \frac{\exp(\pi\sqrt{2N/3})}{4\sqrt{3N}}$ since 1918 by Hardy and Ramanujan [11].

For the asymptotic growth rate of $|D(n)|$, good upper and lower bounds are known. Burns [6] showed there are positive constants c_1 and c_2 such that $4^n/(c_1 n) \leq |D(n)| \leq 4^n/((\log n)^{c_2} \sqrt{n})$ for all sufficiently large n . Ruskey *et al.* [27] presented a fast algorithm to enumerate degree sequences of length n using the “Reverse Search” approach originated by Avis and Fukuda [2]. The algorithm appears to run in constant amortized time (CAT), though no proof has been given yet to the knowledge of the author. No matter whether it is truly CAT or not, it is clear that its time complexity is at least $\Omega(4^n/n)$ based on the lower bound of $|D(n)|$ as proved in [6]. Exact values of $|D(n)|$ up to $n = 31$ were obtained in Iványi *et al.* [14] by testing every sequence in the set $E(n)$ of even bounded sequences of length n (that is, a non-increasing positive integer sequence of length n with even sum and largest term less than n) for its graphicness. A sequence in $E(n)$ can be tested whether it is graphic in linear time and $|E(n)|$ is asymptotically $4^n/(8\sqrt{\pi n})$ [14]. Therefore, this filtering method of calculating $|D(n)|$ has an asymptotic time complexity of $\Theta(\sqrt{n}4^n)$ and space complexity of $O(n)$. By our own testing, the method in [27] appears to be faster than the filtering method in [14] if both are restricted to run in single-threaded mode, which would make sense because the “Reverse Search” based approach would have a time complexity of $O(4^n/((\log n)^{c_2} \sqrt{n}))$ under the assumption that it runs in CAT.

The OEIS entry [A095268](#) records the values of $|D(n)|$. In 2006 Gordon F. Royle posed the question whether $|D(n)|/|D(n-1)|$ tend to the limit 4 when n approaches infinity [29] as there seems to be no apparent reason why $|D(n)|/|D(n-1)|$ would tend to the same limit as $|E(n)|/|E(n-1)|$. In this paper we will present two fast novel algorithms to calculate $|D(n)|$ based on several recurrences, circumventing the need to generate all degree sequences with a given length as performed in [27, 14]. We will show that both of these algorithms run in polynomial time, which appear to be the first such algorithms for this problem and are asymptotically faster than previous exponential algorithms shown above, at the expense of larger space requirement. The numerical results of $|D(n)|$ so obtained up to $n = 118$ tend to agree with the conjecture posed by Royle. Similar techniques are utilized to design

fast polynomial time algorithms to count the number of degree sequences of connected and biconnected graphs on n vertices. The only known previous algorithms to count them seem to be first generating all the degree sequences in $D(n)$ and then counting a subset satisfying the relevant connected and biconnected conditions, which make them all run in exponential time. A related problem of counting the number of degree sequences with fixed degree sum (instead of fixed length) of connected and biconnected *multigraphs* that might contain multiple edges is considered by Rødseth *et al.* [24].

We note that the degree sequences considered in this paper are unordered. For example, the degree sequences 211, 121 and 112 of the path graph with three vertices are all treated as the same. When regarded as ordered degree sequences, they are considered distinct. Ordered partitions are called compositions in the literature [1]. The efficient computation of the number $f(n)$ of ordered degree sequences of length n allowing zero terms (the OEIS entry [A005155](#)) has been solved by Stanley [30] using the theory of zonotopes. For example, the exact value of $f(200)$ is about 9.97×10^{458} . The asymptotic order of $f(n)$ is shown on its OEIS entry by Vaclav Kotesovec to be $\frac{\Gamma(3/4)}{2^{3/4}\sqrt{e\pi}}n^{n-1/4}(1 - \frac{11\pi}{24\sqrt{n}(\Gamma(3/4))^2})$, which is also asymptotic to $\frac{\Gamma(3/4)}{2^{3/4}\sqrt{e\pi}}n^{n-1/4}$. By contrast, the tight asymptotic order of $|D(n)|$ is unknown and we are unable to resolve it at this time.

The rest of the paper is organized as follows. In Section 2, we formalize the necessary definitions, derive the relevant formulae and describe the two algorithms to compute $|D(n)|$. In Section 3 we present algorithms based on similar techniques to compute the number of degree sequences of connected graphs on n vertices. In Section 4 we present an algorithm based on similar dynamic programming techniques and additional combinatorial observations to compute the number of degree sequences of biconnected graphs on n vertices. In Section 5 we analyze the computational complexity of the algorithms. In Section 6 we give some brief computational results that serve as proof of concept. In Section 7 we conclude and suggest further research directions.

2. Description of the Two Algorithms for $|D(n)|$

In this section we first review the definitions needed to describe the two algorithms to calculate $|D(n)|$. We then present the relevant formulae for $|D(n)|$ and the pseudo-codes of our algorithms. For the reader's convenience, the terminology employed in this paper is summarized in Table 1.

2.1. Definitions and the Basic Algorithm

The notion of *partitions* in number theory is standard and can be found in many textbooks, for example [12, 1]. A partition of an integer is a way of writing the integer as a sum of positive integers in which the order of the summands (also called parts) is irrelevant. The parts in a partition are usually written in non-increasing order, for example, $8 = 4 + 3 + 1$. A partition can be visually represented by its Ferrers diagram as a left-aligned array of dots so that the number of dots in the i -th row of the array is equal to the i -th part of the partition. The Durfee square is the upper-left largest square that is embedded within a partition's Ferrers diagram. Figure 1 shows the Ferrers diagram of the example partition $8 = 4 + 3 + 1$. This partition has three parts, the largest part is 4, and its Durfee square contains two rows and two columns.

Table 1: Terminology used in this paper

Term	Meaning
$P(N)$	set of all partitions of an integer N
$P(N, k, l)$	set of partitions of an integer N into at most l parts with largest part at most k
$P(N, k, l, s)$	subset of $P(N, k, l)$ determined by integer s
$P'(N, k, l)$	set of partitions in $P(N, k, l)$ with exactly l parts and largest part exactly k
$P'(N, k, l, s)$	set of partitions in $P(N, k, l, s)$ with exactly l parts and largest part exactly k
$G(N)$	set of all graphical partitions of an even integer N
$G'(N, k, l)$	set of graphical partitions in $P'(N, k, l)$
$G'(N, l)$	set of graphical partitions of N with exactly l parts
$H'(N, l)$	set of graphical partitions of N with exactly l parts and largest part exactly $l - 1$
$L'(N, l)$	set of graphical partitions of N with exactly l parts and largest part less than $l - 1$
$d(\pi)$	size of the Durfee square of the partition π
$r(\pi)$	corank vector of the partition π
$D(n)$	set of (zero-free) degree sequences of length n
$B(n)$	subset of $D(n)$ with largest part exactly $n - 1$ and smallest part exactly 1
$C(n)$	subset of $D(n)$ with smallest part exactly 1
$D_0(n)$	set of degree sequences of length n allowing zero terms
$D_b(n)$	set of degree sequences of biconnected graphs on n vertices
$D_c(n)$	set of degree sequences of connected graphs on n vertices
$D_d(n)$	subset of $D(n)$ that are forcibly disconnected
$E(n)$	set of even bounded sequences of length n
$f(n)$	number of ordered degree sequences of length n allowing zero terms
$H(n)$	subset of $D(n)$ with largest part exactly $n - 1$
$L(n)$	subset of $D(n)$ with largest part less than $n - 1$
$S(n)$	subset of $D(n)$ with largest part exactly $n - 2$
$W(n)$	subset of $D(n)$ with largest part less than $n - 1$ and smallest part exactly 1
$I_e(n)$	$\{N n \leq N \leq n(n - 1), N \text{ is an even integer}\}$
$I'_e(n)$	$\{N 2(n - 1) \leq N \leq n(n - 1), N \text{ is an even integer}\}$
$I''_e(n)$	$\{N 2n - 3 \leq N \leq n(n - 2), N \text{ is an even integer}\}$
$J_e(n)$	$\{N n \leq N \leq n(n - 2), N \text{ is an even integer}\}$
$J'_e(n)$	$\{N n \leq N < n(n - 1)/2, N \text{ is an even integer}\}$

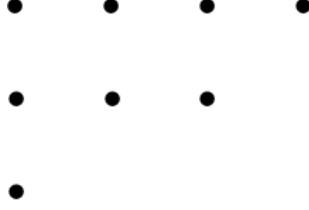


Figure 1: Ferrers diagram of the partition $8=4+3+1$. Its Durfee square size is 2.

Following the notation of Barnes and Savage [3], let $P(N, k, l)$ denote the set of partitions of an integer N into at most l parts with largest part at most k , and for integer s , define $P(N, k, l, s)$ by

$$P(N, k, l, s) = \begin{cases} \emptyset & \text{if } s < 0; \\ \{\pi \in P(N, k, l) \mid s + \sum_{i=1}^j r_i(\pi) \geq j, 1 \leq j \leq d(\pi)\} & \text{if } s \geq 0, \end{cases}$$

where $d(\pi)$ is the size (number of rows) of the Durfee square of the partition π and $r(\pi) = (r_1(\pi), \dots, r_{d(\pi)}(\pi))$ is the corank vector of π with $r_i(\pi) = \pi'_i - \pi_i$ and π_i and π'_i are the number of dots in the i -th row and column of the Ferrers diagram of π , respectively, for $1 \leq i \leq d(\pi)$. The example partition $8 = 4 + 3 + 1$ in Figure 1 has the corank vector $(-1, -1)$.

Additionally, as in [3], define $P'(N, k, l)$ and $P'(N, k, l, s)$ to be the subsets of $P(N, k, l)$ and $P(N, k, l, s)$, respectively, consisting of those partitions with *exactly* l parts and largest part *exactly* k . Define $G'(N, k, l)$ to be the set of graphical partitions in $P'(N, k, l)$. Then we have the following lemma that establishes a relationship between a set of graphical partitions and a set of partitions.

Lemma 1. *For even integer $N \geq 0$, $G'(N, k, l) = P'(N, k, l, 0)$.*

Proof. From the definition of $P'(N, k, l, s)$, we have

$$P'(N, k, l, 0) = \{\pi \in P'(N, k, l) \mid \sum_{i=1}^j r_i(\pi) \geq j, 1 \leq j \leq d(\pi)\}.$$

From the definition of $G'(N, k, l)$, we have

$$G'(N, k, l) = \{\pi \in P'(N, k, l) \mid \pi \text{ is graphical}\}.$$

Now based on the Nash-Williams condition [26, 25, 28], a partition π of an even integer is graphical if and only if $\sum_{i=1}^j r_i(\pi) \geq j, 1 \leq j \leq d(\pi)$. So we have $G'(N, k, l) = P'(N, k, l, 0)$. \square

Before presenting the algorithms for $|D(n)|$ we need the following lemma that connects the function $|P'(*, *, *, *)|$ to its relative $|P(*, *, *, *)|$ and it will be used to bridge the gap between the problem of counting graphical partitions and that of counting partitions. Its proof is based on the observation that there is a bijection between $P'(N, k, l)$ and $P(N - k - l + 1, k - 1, l - 1)$.

Lemma 2. [3, Lemma 4] For $N > 0, 1 \leq k, l \leq N, s \geq 0$,

$$|P'(N, k, l, s)| = |P(N - k - l + 1, k - 1, l - 1, s + l - k - 1)|.$$

The following theorem materializes a connection between particular variants of partition counting functions and graphical partition counting functions. It is important to our algorithms to calculate $|D(n)|$.

Theorem 3. For even integer $N \geq 0$, $|G'(N, k, l)| = |P(N - k - l + 1, k - 1, l - 1, l - k - 1)|$.

Proof. Immediate from Lemma 1 and Lemma 2. \square

There are tons of efficient algorithms to calculate various partition functions in the literature. Our algorithms for $|D(n)|$ need an elegant recurrence of Barnes and Savage [3] that makes it easy to design an efficient dynamic programming algorithm to calculate all $|P(*, *, *, *)|$ values. We now derive the simple connection between $D(n)$ and $G'(N, k, l)$ and then present our basic algorithm to calculate $|D(n)|$ that eventually uses those $|P(*, *, *, *)|$ values.

Recall that $D(n)$ denotes the set of degree sequences of length n . Since each term in a degree sequence of length n is between 1 and $n - 1$ inclusive, it follows that each degree sequence in $D(n)$ can be seen as a graphical partition of some even integer N with exactly n parts and largest part exactly k where $n \leq N \leq n(n - 1)$ and $1 \leq k \leq n - 1$. Define $G'(N, l)$ to be the set of graphical partitions of N with *exactly* l parts. Also define an index set $I_e(n) = \{N | n \leq N \leq n(n - 1), N \text{ is an even integer}\}$. Based on the above observations we clearly have

$$D(n) = \bigcup_{N \in I_e(n)} G'(N, n) \tag{1}$$

and

$$G'(N, n) = \bigcup_{k=1}^{n-1} G'(N, k, n), \tag{2}$$

which can be combined into

$$D(n) = \bigcup_{N \in I_e(n)} \bigcup_{k=1}^{n-1} G'(N, k, n). \tag{3}$$

These are all disjoint unions so that we have

$$|D(n)| = \sum_{N \in I_e(n)} |G'(N, n)|, \tag{4}$$

$$|G'(N, n)| = \sum_{k=1}^{n-1} |G'(N, k, n)|, \tag{5}$$

$$|D(n)| = \sum_{N \in I_e(n)} \sum_{k=1}^{n-1} |G'(N, k, n)|. \tag{6}$$

Substituting the formula in Theorem 3 into Equation (5) and Equation (6) we get

$$|G'(N, n)| = \sum_{k=1}^{n-1} |P(N - k - n + 1, k - 1, n - 1, n - k - 1)|, \quad (7)$$

$$|D(n)| = \sum_{N \in I_e(n)} \sum_{k=1}^{n-1} |P(N - k - n + 1, k - 1, n - 1, n - k - 1)|. \quad (8)$$

With Equation (8) it is sufficient to calculate a finite number of $|P(*, *, *, *)|$ values to compute $|D(n)|$. As mentioned above, all these $|P(*, *, *, *)|$ values can be calculated with a recurrence of Barnes and Savage in the following theorem:

Theorem 4. [3, Theorem 1] $|P(N, k, l, s)|$ is defined by:

$$\begin{aligned} |P(N, k, l, s)| := & \\ & \text{if}((N < 0) \text{ or } (k < 0) \text{ or } (l < 0) \text{ or } (s < 0)) \quad \text{then : 0} \\ & \text{else if}(N = 0) \quad \text{then : 1} \\ & \text{else if}((k = 0) \text{ or } (l = 0)) \quad \text{then : 0} \\ & \text{else if}(k > N) \quad \text{then : } |P(N, N, l, s)| \\ & \text{else if}(l > N) \quad \text{then : } |P(N, k, N, s)| \\ & \text{else if}(s > N) \quad \text{then : } |P(N, k, l, N)| \\ & \text{else : } |P(N, k - 1, l, s)| + |P(N, k, l - 1, s)| - |P(N, k - 1, l - 1, s)| \\ & \quad + |P(N - k - l + 1, k - 1, l - 1, s + l - k - 1)| \end{aligned}$$

Algorithm 1: Basic Algorithm to compute $|D(n)|$. The index set $I_e(n) = \{N | n \leq N \leq n(n - 1), N \text{ is an even integer}\}$.

Input: A positive integer n
Output: $|D(n)|$

- 1 $N \leftarrow n(n - 1)$
- 2 Allocate a four dimensional array $P[N - n + 1][n - 1][n][N]$
- 3 Fill in the array P using dynamic programming based on Theorem 4
- 4 $S \leftarrow 0$
- 5 **for** $i \in I_e(n)$ **do**
- 6 **for** $j \leftarrow 1$ **to** $\min(n - 1, i - n + 1)$ **do**
- 7 $S \leftarrow S + P[i - j - n + 1][j - 1][n - 1][n - j - 1]$
- 8 **return** S

With the above definitions and techniques, we present our basic algorithm to calculate $|D(n)|$ as pseudo-codes in Algorithm 1. The declaration of the array P in line 2 specifies the sizes of the four dimensions. When filling in and accessing the array elements we use the convention that array indices start from 0. Also note that based on the recurrence in

Theorem 4 the $|P(*, *, l, *)|$ values only depend on the $|P(*, *, l - 1, *)|$ values, and for the purpose of calculating $|D(n)|$ only the $|P(*, *, n - 1, *)|$ values are used. These make it feasible to allocate only size 2 for the third dimension of the array P to save memory. For each fixed i in line 5, the sum accumulated in the inner loop from lines 6 to 7 is the value $|G'(i, n)|$ based on Equation (7) and it can also be saved as additional output if desired. The integer sequence $|D(n)|$ is in the OEIS entry [A095268](#).

We note that Equation (8) and Algorithm 1 are obtained by partitioning the set $D(n)$ into disjoint subsets $G'(N, n)$, $N \in I_e(n)$ based on degree sums. Alternatively, we can partition the set $D(n)$ into disjoint subsets based on the largest term in each degree sequence. This way effectively only produces a formula that interchanges the two sum symbols in the double sums in Equation (6) and Equation (8). The change that can be made to Algorithm 1 based on the equivalent new formula is to simply interchange the two for loop lines 5 and 6, making the sums accumulated in the inner loop the number of degree sequences of given length n and fixed largest term j . Thus, we can see that the set of degree sequences of fixed length and fixed sum and the set of degree sequences of fixed length and fixed largest term can all be efficiently counted.

2.2. Some Novel Recurrences and the Improved Algorithm

In this section we first review some old and also derive novel recurrences involving the sizes of $D(n)$ and various related sets. We then present an improved algorithm to compute $|D(n)|$ based on these recurrences.

Define $H(n)$ to be the set of degree sequences of length n with largest part exactly $n - 1$ and $L(n)$ the set of degree sequences of length n with largest part less than $n - 1$. Clearly $D(n)$ is the disjoint union of $H(n)$ and $L(n)$. Let $D_0(n)$ denote the set of degree sequences of length n allowing zero terms. The OEIS entry [A004251](#) records the values of $|D_0(n)|$. We now present two propositions demonstrating the relationships among the cardinalities of these sets that will be used by our improved algorithm. The following proposition is not hard to derive by the simple observation that $D_0(n)$ is the disjoint union of $D(n)$ and the set of degree sequences of length n containing at least one zero.

Proposition 5. [14, Lemma 2]: $|D_0(n)| = |D_0(n - 1)| + |D(n)|$, for $n \geq 2$. And

$$|D_0(n)| = \begin{cases} 1 & \text{if } n = 1; \\ 1 + \sum_{i=2}^n |D(i)| & \text{if } n \geq 2. \end{cases}$$

Based on Proposition 5 it is a trivial thing to compute $|D_0(n)|$ once all $|D(i)|$ such that $i \leq n$ have been computed. We also have the following proposition that transforms the problem of computing $|D(n)|$ to that of computing $|L(n)|$.

Proposition 6. $|H(n)| = |D_0(n - 1)|$, and $|D(n)| = |D_0(n - 1)| + |L(n)|$, for $n \geq 2$.

Proof. Each simple undirected graph G with n vertices having a vertex of degree $n - 1$ can be transformed into a graph G_1 of $n - 1$ vertices (possibly with isolated vertices) by removing the vertex of degree $n - 1$. Thus, $H(n) \subseteq D_0(n)$.

Conversely, each graph G_1 of $n - 1$ vertices (possibly with isolated vertices) can be transformed into a graph G with n vertices having a vertex of degree $n - 1$ by adding a new

vertex and making it adjacent to all the vertices originally in G_1 . Thus, $D_0(n) \subseteq H(n)$. This bijection establishes that $|H(n)| = |D_0(n-1)|$.

Since $D(n) = H(n) \cup L(n)$, and this is a disjoint union, we have $|D(n)| = |H(n)| + |L(n)| = |D_0(n-1)| + |L(n)|$. \square

By Propositions 5 and 6, it is sufficient to compute $|L(n)|$ in order to compute $|D(n)|$ when all $|D(i)|$ such that $i < n$ are already known. Not surprisingly, we will show that $|L(n)|$ can also be computed eventually through the $|P(*, *, *, *)|$ values.

Define $L'(N, l)$ to be the set of graphical partitions of N with exactly l parts and largest part less than $l-1$. Also define an index set $J_e(n) = \{N | n \leq N \leq n(n-2), N \text{ is an even integer}\}$. It is easily seen that

$$L(n) = \bigcup_{N \in J_e(n)} L'(N, n) \quad (9)$$

and this is a disjoint union so that we have

$$|L(n)| = \sum_{N \in J_e(n)} |L'(N, n)|. \quad (10)$$

By definition we have

$$L'(N, n) = \bigcup_{k=1}^{n-2} G'(N, k, n) \quad (11)$$

and by disjointness of this union we have

$$|L'(N, n)| = \sum_{k=1}^{n-2} |G'(N, k, n)|. \quad (12)$$

Combining Equations (10) and (12) and substituting the formula in Theorem 3 into Equation (12) we get

$$\begin{aligned} |L(n)| &= \sum_{N \in J_e(n)} \sum_{k=1}^{n-2} |G'(N, k, n)| \\ &= \sum_{N \in J_e(n)} \sum_{k=1}^{n-2} |P(N-k-n+1, k-1, n-1, n-k-1)| \end{aligned} \quad (13)$$

and this shows how $|L(n)|$ can also be calculated by summing a finite number of $|P(*, *, *, *)|$ values. In fact, the calculation of $|L(n)|$ can be further sped up by the following proposition.

Proposition 7. $|L'(N, n)| = |L'(n(n-1) - N, n)|$, for $N \in J_e(n)$.

Proof. Each degree sequence $d_1 \geq d_2 \geq \dots \geq d_n > 0$ in $L(n)$ with sum N and $d_1 < n-1$ can be transformed into a degree sequence $n-1-d_n \geq n-1-d_{n-1} \geq \dots \geq n-1-d_1 > 0$ with sum $n(n-1) - N$, which still lies in $L(n)$ since $n-1-d_n < n-1$. \square

Algorithm 2: Improved Algorithm to compute $|D(n)|$, assuming all $|D(i)|$ such that $i < n$ are already computed. The index set $J'_e(n) = \{N | n \leq N < n(n-1)/2, N \text{ is an even integer}\}$.

Input: A positive integer n

Output: $|D(n)|$

```

1  $N \leftarrow n(n-1)/2$ 
2 Allocate a four dimensional array  $P[N-n+1][n-2][n][N]$ 
3 Fill in the array  $P$  using dynamic programming based on Theorem 4
4  $S \leftarrow 0$ 
5 for  $i \in J'_e(n)$  do
6   for  $j \leftarrow 1$  to  $\min(n-2, i-n+1)$  do
7      $S \leftarrow S + P[i-j-n+1][j-1][n-1][n-j-1]$ 
8  $S \leftarrow 2S$ 
9 if  $N$  is even then
10  for  $j \leftarrow 1$  to  $\min(n-2, N-n+1)$  do
11     $S \leftarrow S + P[N-j-n+1][j-1][n-1][n-j-1]$ 
12 Calculate  $|D_0(n-1)|$  using saved values of  $|D(i)|, i < n$  by Proposition 5
13 return  $S + |D_0(n-1)|$ 

```

Based on Equation (10) and Proposition 7, it is sufficient to calculate the $|L'(N, n)|$ values for even N between n and $n(n-1)/2$ in order to calculate $|L(n)|$, apparently saving the work load by a constant factor.

We now present an improved Algorithm 2 to calculate $|D(n)|$ based on the above definitions and techniques, assuming all values of $|D(i)|$ such that $i < n$ have been calculated. The index set $J'_e(n)$ used in the improved algorithm is defined as $J'_e(n) = \{N | n \leq N < n(n-1)/2, N \text{ is an even integer}\}$. The variable S is used to store the value of $|L(n)|$. As in Algorithm 1, the size for the third dimension of the array P can also be reduced to 2 for the purpose of computing $|L(n)|$ and $|D(n)|$. The individual sums accumulated in the loop from lines 6 to 7 and lines 10 to 11 are the values $|L'(*, n)|$ based on Equation (12) and Equation (13) and they together with the $|L'(n(n-1) - *, n)|$ values can also be saved as additional output if desired. Algorithm 2 is called an improved algorithm because it achieves a constant factor improvement over Algorithm 1 (see complexity analysis in Section 5). Although using more time and space, Algorithm 1 does have the advantage of applicability for any n without the need to know all the values of $|D(i)|$ such that $i < n$ as in Algorithm 2.

Before concluding this section we derive one more interesting recurrence. Define $H'(N, l)$ to be the set of graphical partitions of N with exactly l parts and largest part exactly $l-1$. Also define an index set $I'_e(n) = \{N | 2(n-1) \leq N \leq n(n-1), N \text{ is an even integer}\}$. By definitions it is easily seen that

$$H(n) = \bigcup_{N \in I'_e(n)} H'(N, n) \quad (14)$$

and this is a disjoint union so that we have

$$|H(n)| = \sum_{N \in I'_e(n)} |H'(N, n)|. \quad (15)$$

By definition we have

$$H'(N, n) = G'(N, n - 1, n), \quad (16)$$

and

$$G'(N, n) = H'(N, n) \cup L'(N, n), \quad |G'(N, n)| = |H'(N, n)| + |L'(N, n)|. \quad (17)$$

The following proposition indicates a symmetry of $|H'(N, n)|$ similar to that of $|L'(N, n)|$ as shown in Proposition 7.

Proposition 8. $|H'(N, n)| = |H'((n + 2)(n - 1) - N, n)|$, for $N \in I'_e(n)$.

Proof. Each degree sequence $d_1 \geq d_2 \geq \dots \geq d_n > 0$ in $H(n)$ with sum N and $d_1 = n - 1$ can be first transformed into a degree sequence allowing zero terms (of length n , by appending a zero after $d_n - 1$) $d_2 - 1 \geq d_3 - 1 \geq \dots \geq d_n - 1 \geq 0 \geq 0$ by applying the Havel-Hakimi algorithm [13, 9] which characterizes equivalent realizable sequences, and then transformed into a degree sequence (of length n) $n - 1 \geq n - d_n \geq n - d_{n-1} \geq \dots \geq n - d_2 > 0$ with sum $(n + 2)(n - 1) - N$, which still lies in $H(n)$. \square

From available numerical results we observe that for fixed n the sequence of $|H'(N, n)|$ for $N \in I'_e(n)$ is unimodal and the maximum value in the sequence occurs when N is an even integer closest to $(n^2 + n - 2)/2$. It is also observed that for fixed n the sequence of $|L'(N, n)|$ for $N \in J_e(n)$ is unimodal and the maximum value in the sequence occurs when N is an even integer closest to $(n^2 - n)/2$. Based on Equation (17) and available numerical results we also conjecture that for fixed n the sequence of $|G'(N, n)|$ for $N \in I_e(n)$ is unimodal and the maximum value in the sequence occurs when N is closest to $n^2/2 - n/6$. One of the ways to determine the asymptotic order of $|D(n)|$ might be to determine the order of $|G'(N, n)|$ for a narrow range of N centered around $n^2/2 - n/6$. The rigorous proof of these results might need advanced techniques.

3. Algorithms to compute the number of degree sequences of connected graphs on n vertices

In this section we apply the techniques employed in the two algorithms for computing $|D(n)|$ in Section 2 to derive similar algorithms for computing the number of degree sequences of connected graphs on n vertices.

Recall that a degree sequence is called *potentially* connected (resp. disconnected) if it has a connected (resp. disconnected) realization and it is called *forcibly* connected (resp. disconnected) if all of its realizations are connected (resp. disconnected). Let $D_c(n)$ denote the subset of $D(n)$ that are potentially connected and $D_d(n)$ the subset of $D(n)$ that are forcibly disconnected. Clearly $D(n)$ is the disjoint union of $D_c(n)$ and $D_d(n)$. A degree

sequence $d = (d_1, d_2, \dots, d_n)$ in $D(n)$ is potentially connected if and only if $\sum_{i=1}^n d_i \geq 2(n-1)$ [5]. Because of this simple characterization it is easily seen by definitions that

$$D_c(n) = \bigcup_{N \in I'_e(n)} G'(N, n) = \bigcup_{N \in I'_e(n)} \bigcup_{k=1}^{n-1} G'(N, k, n), \quad (18)$$

and

$$D_d(n) = \bigcup_{N \in I_e(n) - I'_e(n)} G'(N, n) = \bigcup_{N \in I_e(n) - I'_e(n)} \bigcup_{k=1}^{n-1} G'(N, k, n). \quad (19)$$

Again by disjointness we have

$$\begin{aligned} |D_c(n)| &= \sum_{N \in I'_e(n)} \sum_{k=1}^{n-1} |G'(N, k, n)| \\ &= \sum_{N \in I'_e(n)} \sum_{k=1}^{n-1} |P(N - k - n + 1, k - 1, n - 1, n - k - 1)| \end{aligned} \quad (20)$$

and

$$\begin{aligned} |D_d(n)| &= \sum_{N \in I_e(n) - I'_e(n)} \sum_{k=1}^{n-1} |G'(N, k, n)| \\ &= \sum_{N \in I_e(n) - I'_e(n)} \sum_{k=1}^{n-1} |P(N - k - n + 1, k - 1, n - 1, n - k - 1)|. \end{aligned} \quad (21)$$

If $|D(n)|$ has already been obtained, we can choose to compute $|D_d(n)|$ in order to get $|D_c(n)| = |D(n)| - |D_d(n)|$ for performance reasons. Now we present Algorithm 3 to compute $|D_c(n)|$ directly based on Equation (20) and Algorithm 4 to compute $|D_c(n)|$ indirectly through computing $|D_d(n)|$ based on Equation (21). As in Algorithms 1 and 2 we can still allocate size 2 for the third dimension of the array P .

The OEIS entry [A007721](#) records the values of $|D_c(n)|$.

4. Algorithm to compute the number of degree sequences of biconnected graphs on n vertices

In this section we will derive Algorithm 5 to count the set $D_b(n)$ of degree sequences of biconnected graphs on n vertices. A degree sequence $d_1 \geq d_2 \geq \dots \geq d_n$ in $D(n)$ is potentially biconnected if and only if $\sum_{i=1}^n d_i \geq 2n - 4 + 2d_1$ and $d_n \geq 2$ [31]. Since this characterization expresses the lower bound of the degree sum as a quantity that can vary due to d_1 , there is not a simple algorithm like Algorithm 3 or Algorithm 4 to count $|D_b(n)|$. However, we will show that $|D_b(n)|$ can still be computed eventually through the $|P(*, *, *, *)|$ values and the unrestricted partition function $|P(*)|$ values.

Algorithm 3: Algorithm to compute $|D_c(n)|$ directly. The index set $I'_e(n) = \{N | 2(n-1) \leq N \leq n(n-1), N \text{ is an even integer}\}$.

Input: A positive integer n
Output: $|D_c(n)|$

- 1 $N \leftarrow n(n-1)$
- 2 Allocate a four dimensional array $P[N-n+1][n-1][n][N]$
- 3 Fill in the array P using dynamic programming based on Theorem 4
- 4 $S \leftarrow 0$
- 5 **for** $i \in I'_e(n)$ **do**
- 6 **for** $j \leftarrow 1$ **to** $\min(n-1, i-n+1)$ **do**
- 7 $S \leftarrow S + P[i-j-n+1][j-1][n-1][n-j-1]$
- 8 **return** S

Algorithm 4: Algorithm to compute $|D_c(n)|$ indirectly, assuming $|D(n)|$ is known. The index set $I_e(n) - I'_e(n) = \{N | n \leq N < 2(n-1), N \text{ is an even integer}\}$.

Input: A positive integer n
Output: $|D_c(n)|$

- 1 $N \leftarrow 2(n-2)$
- 2 Allocate a four dimensional array $P[N-n+1][n-1][n][N]$
- 3 Fill in the array P using dynamic programming based on Theorem 4
- 4 $S \leftarrow 0$
- 5 **for** $i \in I_e(n) - I'_e(n)$ **do**
- 6 **for** $j \leftarrow 1$ **to** $\min(n-1, i-n+1)$ **do**
- 7 $S \leftarrow S + P[i-j-n+1][j-1][n-1][n-j-1]$
- 8 **return** $|D(n)| - S$

Define $S(n)$ to be the set of degree sequences of length n with largest part exactly $n-2$, $C(n)$ the set of degree sequences of length n with smallest part exactly 1 and $B(n)$ the set of degree sequences of length n with largest part exactly $n-1$ and smallest part exactly 1. Then, $D_2(n) = D(n) - C(n)$ is the set of degree sequences of length n with smallest part at least 2. We next establish some connections between these sets and show that $|D_2(n)|$ can be efficiently computed.

Proposition 9. $|C(n)| = |B(n)| + |S(n)|, |B(n)| = |D_0(n-2)|$, for $n \geq 3$.

Proof. Notice that $C(n)$ is the disjoint union of $B(n)$ and the set $W(n)$ of degree sequences of length n with largest part less than $n-1$ and smallest part exactly 1, so we have $|C(n)| = |B(n)| + |W(n)|$. Each degree sequence $d_1 \geq d_2 \geq \dots \geq d_n$ in $W(n)$ can be transformed into a degree sequence $n-1-d_n \geq n-1-d_{n-1} \geq \dots \geq n-1-d_1$, which is a degree sequence in $S(n)$ since $d_1 < n-1$ and $d_n = 1$. Conversely, each degree sequence in $S(n)$ can be transformed into one in $W(n)$ in a similar way. This bijection establishes that $|W(n)| = |S(n)|$, and then the first equation follows.

Each simple undirected graph G with n vertices having a vertex of degree $n - 1$ and a vertex of degree 1 can be transformed into a graph G_2 of $n - 2$ vertices (possibly with isolated vertices) by first removing the vertex of degree $n - 1$, resulting in an isolated vertex (previously with degree 1), and then removing the newly appeared isolated vertex. Conversely, each graph G_2 of $n - 2$ vertices (possibly with isolated vertices) can be transformed to a graph G with n vertices having a vertex of degree $n - 1$ and a vertex of degree 1 by first adding a new isolated vertex and then adding a second vertex and making it adjacent to all the other $n - 1$ vertices. This bijection establishes that $|B(n)| = |D_0(n - 2)|$. \square

Now $|B(n)|$ can be efficiently computed based on Proposition 9 and Proposition 5. As noted at the end of Section 2.1, $|S(n)|$ can be efficiently computed. Each degree sequence in $S(n)$ must have sum N such that $2n - 3 \leq N \leq n(n - 2)$. Define an index set $I_e''(n) = \{N | 2n - 3 \leq N \leq n(n - 2), N \text{ is an even integer}\}$. Clearly,

$$S(n) = \bigcup_{N \in I_e''(n)} G'(N, n - 2, n).$$

By disjointness, we have

$$|S(n)| = \sum_{N \in I_e''(n)} |G'(N, n - 2, n)| = \sum_{N \in I_e''(n)} |P(N - 2n + 3, n - 3, n - 1, 1)|. \quad (22)$$

Thus, $|C(n)|$ can be efficiently computed based on Proposition 9 and we can therefore efficiently compute $|D_2(n)|$ as

$$|D_2(n)| = |D(n)| - |C(n)|. \quad (23)$$

Next we will show that the combinatorial trick of expressing $|D_2(n)|$ in a different way allows us to compute $|D_b(n)|$ efficiently.

Recall that $D_b(n)$ is the set of degree sequences of biconnected graphs on n vertices. Clearly $D_b(n) \subseteq D_2(n)$ based on the characterization of potentially biconnected degree sequences [31]. Denote $D_{2 \setminus b}(n) = D_2(n) - D_b(n)$, i.e., the set of those degree sequences $d_1 \geq d_2 \geq \dots \geq d_n$ such that $\sum_{i=1}^n d_i < 2n - 4 + 2d_1$ and $d_n \geq 2$. Clearly, we can express $|D_2(n)|$ in a second way as

$$|D_2(n)| = |D_b(n)| + |D_{2 \setminus b}(n)|. \quad (24)$$

This fulfills our goal of efficiently computing $|D_b(n)| = |D_2(n)| - |D_{2 \setminus b}(n)|$ as long as $|D_{2 \setminus b}(n)|$ can be computed efficiently, which will be shown next.

Consider the possible degree sequences $d_1 \geq d_2 \geq \dots \geq d_n$ in $D_{2 \setminus b}(n)$. Since $D_{2 \setminus b}(n) \subseteq D_2(n)$, $d_1 \geq d_n \geq 2$. The possible choices for d_1 are $d_1 = 2, \dots, n - 1$ and they are considered individually below.

1. $d_1 = 2$: It is required that $\sum_{i=1}^n d_i < 2n$. This is impossible since we also need $\sum_{i=1}^n d_i \geq 2(n - 1) + 2 = 2n$.
2. $d_1 = 3$: It is required that $\sum_{i=1}^n d_i < 2n + 2$. Again impossible since we also need $\sum_{i=1}^n d_i \geq 2(n - 1) + 3 = 2n + 1$.

3. $d_1 = 2k \geq 4$: It is required that $\sum_{i=1}^n d_i < 2n + 4k - 4$. We also need $\sum_{i=1}^n d_i \geq 2(n-1) + 2k = 2n + 2k - 2$. Thus $\sum_{i=1}^n d_i = 2n + 2k - 2, 2n + 2k, \dots, 2n + 4k - 6$ are all the feasible degree sums. For each feasible degree sum $2n + 2k - 2 + N$, $N = 0, 2, \dots, 2k - 4$, consider the possible degree sequences $d_1 \geq d_2 \geq \dots \geq d_n$ such that $d_1 = 2k$, $d_n \geq 2$, $\sum_{i=1}^n d_i = 2n + 2k - 2 + N$. Clearly the number of possible degree sequences is the number of ways to partition N into $t \leq n - 1$ parts $N = N_1 + N_2 + \dots + N_t$ such that $d_1 \geq 2 + N_1 \geq 2 + N_2 \geq \dots \geq 2 + N_t \geq 2 \geq \dots \geq 2$ is a graphical degree sequence. The particular case $t = 1$ gives the integer sequence $d_1 \geq 2 + N \geq 2 \geq \dots \geq 2$. It is easy to check based on the Nash-Williams condition [26, 25, 28] that for each $N = 0, 2, \dots, 2k - 4$, the integer sequence $d_1 \geq 2 + N \geq 2 \geq \dots \geq 2$ is graphical. For each fixed $N \in \{0, 2, \dots, 2k - 4\}$ and any partition $N = N_1 + N_2 + \dots + N_t$, the integer sequence $l_1 = (d_1, 2 + N, 2, \dots, 2)$ majorizes the integer sequence $l_2 = (d_1, 2 + N_1, 2 + N_2, \dots, 2 + N_t, 2, \dots, 2)$ and since l_1 is graphical, l_2 is also graphical [18, Corollary 3.1.4]. This produces a total of $|P(0)| + |P(2)| + |P(4)| + \dots + |P(2k - 4)|$ possible degree sequences.
4. $d_1 = 2k + 1 \geq 5$: It is required that $\sum_{i=1}^n d_i < 2n + 4k - 2$. We also need $\sum_{i=1}^n d_i \geq 2(n-1) + 2k + 1 = 2n + 2k - 1$. Thus $\sum_{i=1}^n d_i = 2n + 2k, 2n + 2k + 2, \dots, 2n + 4k - 4$ are all the feasible degree sums. For each feasible degree sum $2n + 2k - 1 + N$, $N = 1, 3, \dots, 2k - 3$, consider the possible degree sequences $d_1 \geq d_2 \geq \dots \geq d_n$ such that $d_1 = 2k + 1$, $d_n \geq 2$, $\sum_{i=1}^n d_i = 2n + 2k - 1 + N$. Clearly the number of possible degree sequences is the number of ways to partition N into $t \leq n - 1$ parts $N = N_1 + N_2 + \dots + N_t$ such that $d_1 \geq 2 + N_1 \geq 2 + N_2 \geq \dots \geq 2 + N_t \geq 2 \geq \dots \geq 2$ is a graphical degree sequence. The particular case $t = 1$ gives the integer sequence $d_1 \geq 2 + N \geq 2 \geq \dots \geq 2$. It is easy to check based on the Nash-Williams condition [26, 25, 28] that for each $N = 1, 3, \dots, 2k - 3$, the integer sequence $d_1 \geq 2 + N \geq 2 \geq \dots \geq 2$ is graphical. For each fixed $N \in \{1, 3, \dots, 2k - 3\}$ and any partition $N = N_1 + N_2 + \dots + N_t$, the integer sequence $l_1 = (d_1, 2 + N, 2, \dots, 2)$ majorizes the integer sequence $l_2 = (d_1, 2 + N_1, 2 + N_2, \dots, 2 + N_t, 2, \dots, 2)$ and since l_1 is graphical, l_2 is also graphical [18, Corollary 3.1.4]. This produces a total of $|P(1)| + |P(3)| + |P(5)| + \dots + |P(2k - 3)|$ possible degree sequences.

Based on the above discussions and techniques, we present Algorithm 5 to compute the number of potentially biconnected degree sequences of length n . The variable S is supposed to store the value $|D_{2 \setminus b}(n)|$. The OEIS entry [A007722](#) records the $|D_b(n)|$ values.

Before we conclude this section, we briefly mention that the method in this section can be extended to count the number of degree sequences of triconnected graphs and beyond. Characterization of potentially k -connected degree sequences of length n can be found in Wang and Cleitman [31]. In order to use this characterization, we can still count the set $D_k(n)$ of degree sequences of length n with smallest part at least k in two different ways. Let $C_k(n)$ denote the set of degree sequences of length n with smallest part exactly k . Clearly $|D_k(n)| = |D(n)| - \sum_{i=1}^{k-1} |C_i(n)|$. The first way expresses $|C_i(n)|$ as the sum of the number of degree sequences with fixed largest part $n - i - 1$ and lengths from $n - i$ to n so that $|D_k(n)|$ can be efficiently computed. The second way expresses $|D_k(n)|$ as the sum of the number $|D_{kc}(n)|$ of potentially k -connected degree sequences and the rest $|R(n)|$. Then $|D_{kc}(n)|$ can be efficiently computed by combining the results of computing $|D_k(n)|$ in the two ways.

Algorithm 5: Algorithm to compute $|D_b(n)|$, assuming all $|D(i)|$ such that $i \leq n$ have been computed.

Input: A positive integer $n \geq 5$
Output: $|D_b(n)|$

- 1 Compute $|S(n)|$ using dynamic programming based on Equation (22)
- 2 Compute $|D_0(n-2)|$ based on Proposition 5 using the saved $|D(i)|$ values
- 3 Compute $|C(n)| = |D_0(n-2)| + |S(n)|$ based on Proposition 9
- 4 Compute $|D_2(n)| = |D(n)| - |C(n)|$ using the saved $|D(n)|$ value
- 5 $S \leftarrow 0$
- 6 **for** $i \leftarrow 4$ **to** $n-1$ **do**
- 7 $k \leftarrow \lfloor \frac{i}{2} \rfloor$
- 8 **if** i *is even* **then**
- 9 **for** $j \leftarrow 0$ **to** $2k-4$ *step 2* **do**
- 10 $S \leftarrow S + |P(j)|$
- 11 **else**
- 12 **for** $j \leftarrow 1$ **to** $2k-3$ *step 2* **do**
- 13 $S \leftarrow S + |P(j)|$
- 14 **return** $|D_2(n)| - S$

Counting $R(n)$ in $D_k(n)$ involves an enumeration of all possible values of d_1, \dots, d_{k-1} and will become the most tedious part of the algorithm.

5. Complexity Analysis

As mentioned in Section 1, the previous best known methods to calculate $|D(n)|$ have a time complexity of $\Omega(4^n/n)$ as in [27] and a time complexity of $O(\sqrt{n}4^n)$ and space complexity of $O(n)$ as in [14] respectively. Their exponential running time comes from the fact that they compute $|D(n)|$ by generating all the sequences in $D(n)$. The variants of these algorithms to compute $|D_c(n)|$ and $|D_b(n)|$ by incorporating the relevant tests whether the degree sequence is potentially connected or potentially biconnected have similar exponential time complexity since the tests only add linear extra time for each generated degree sequence. Thus it is infeasible to calculate $|D(n)|$, $|D_c(n)|$ and $|D_b(n)|$ using these methods for n beyond 35. In this section we analyze the computational complexity of the presented algorithms. A summary of the results is presented in Table 2.

We first analyze the complexity of the basic Algorithm 1. The first and fourth dimensions of the array P have sizes $O(n^2)$, the second dimension $O(n)$ and the third dimension a constant as mentioned above. Thus, the space complexity is $O(n^5)$. The dynamic programming step in line 3 costs time $O(n^5)$ as a result. The loop from lines 5 to 7 costs time $O(n^2n) = O(n^3)$, which is dominated by the step in line 3. So Algorithm 1 has time complexity $O(n^5)$.

Similar analysis shows that the improved Algorithm 2 for $|D(n)|$ and Algorithm 3 for $|D_c(n)|$ also have the asymptotic time and space complexity $O(n^5)$. For Algorithm 2, since

Table 2: Complexity of the algorithms

Algorithm	Time	Space
Ruskey <i>et al.</i> [27]	$\Omega(4^n/n)$	$O(n)$
Iványi <i>et al.</i> [14]	$\Theta(\sqrt{n}4^n)$	$O(n)$
Algorithm 1	$O(n^5)$	$O(n^5)$
Algorithm 2	$O(n^5)$	$O(n^5)$
Algorithm 3	$O(n^5)$	$O(n^5)$
Algorithm 4	$O(n^3)$	$O(n^3)$
Algorithm 5	$O(n^5)$	$O(n^5)$

the first and fourth dimensions of the array P have sizes about half of those in the basic Algorithm 1, it achieves an improvement of a constant factor of approximately 4. Although it is not asymptotically faster than the basic algorithm, it does allow us to compute more $|D(n)|$ values since for large values of n the memory issue is more severe than the time issue for these algorithms. Algorithm 4 has time and space complexity $O(n^3)$ due to the reduction of the sizes of the first and fourth dimension of the array P to $O(n)$.

For Algorithm 5, line 1 takes $O(n^5)$ time and $O(n^5)$ space. Lines 2 to 4 take time $O(n)$ using saved values of $|D(i)|, i \leq n$. For lines 5 to 13, all $|P(j)|, j \leq n-5$ can be precomputed in time $O(n^{3/2+\epsilon})$ since $|P(n)|$ can be computed in time $O(n^{1/2+\epsilon})$ [15]. The double for loop from lines 6 to 13 takes time $O(n^2)$. The overall time and space complexity are both $O(n^5)$.

6. Computational Experiences

We note that our calculation of $|D(30)|$ using an optimized C implementation of the basic Algorithm 1 takes a single Intel Core i7 computer with 16 GB memory 1.8 seconds. The improved Algorithm 2 uses 0.7 seconds to get the same result. In contrast, the previous result of [14] estimated that computing $|D(30)|$ using their method would cost a single computer about 72 years. The computation of $|D(118)|$ using the improved Algorithm 2 on a computer with 256 GB memory takes about 2.5 hours.

For $n < 30$ our results of $|D(n)|$ agree with those in [14] while the values of $|D(30)|$ and $|D(31)|$ we calculated are slightly different. From the available numerical results it appears that $|D(n)|/|D(n-1)|$ is slowly increasing when $n \geq 9$ and approaching 4, giving more supporting evidence of Royle's conjecture. We also computed the $|D_0(n)|, |D_c(n)|$ and $|D_b(n)|$ values and incorporated all available numerical results into their respective OEIS entries.

During the preparation of this manuscript, the author became aware of the results in [17], which calculated the values of $|D(n)|$ up to $n = 34$. The method employed therein seemed more complicated and not formally published to the knowledge of the author.

7. Conclusions

In this paper we presented dynamic programming algorithms to compute the number of degree sequences of simple graphs, simple connected graphs and simple biconnected graphs on n vertices exactly all based on a known recurrence of Barnes and Savage. They are the

first known polynomial time algorithms to compute these functions and are asymptotically much faster than previous best known counterparts. Using these fast algorithms we extend the known exact values of $|D(n)|$, $|D_0(n)|$, $|D_c(n)|$ and $|D_b(n)|$ up to $n = 118$ and these results have been included in their OEIS entries. Further research include how to determine the tight asymptotic order of these functions, how to efficiently compute the number of forcibly k -connected degree sequences of length n and determine its asymptotic order.

8. Acknowledgements

This research has been supported by the Startup Fund of Georgia Southern University.

References

- [1] George E. Andrews. *The Theory of Partitions*. Cambridge University Press, 1984.
- [2] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(13):21–46, 1996. First International Colloquium on Graphs and Optimization.
- [3] Tiffany M. Barnes and Carla D. Savage. A recurrence for counting graphical partitions. *Electronic Journal of Combinatorics*, 2, 1995.
- [4] Tiffany M. Barnes and Carla D. Savage. Efficient generation of graphical partitions. *Discrete Applied Mathematics*, 78(13):17–26, 1997.
- [5] Claude Berge. *Graphs and Hypergraphs*. American Elsevier Pub. Co, 1973.
- [6] Jason Matthew Burns. *The number of degree sequences of graphs*. PhD thesis, Massachusetts Institute of Technology. Dept. of Mathematics, 2007.
- [7] P. Erdős and T. Gallai. Graphs with given degree of vertices. *Mat. Lapok*, 11:264–274, 1960.
- [8] P. Erdős and L. B. Richmond. On graphical partitions. *Combinatorica*, 13(1):57–63, 1993.
- [9] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.
- [10] Frank Harary and Edgar M. Palmer. *Graphical Enumeration*. Academic Press, 1973.
- [11] G. H. Hardy and S. Ramanujan. Asymptotic formulae in combinatory analysis. *Proc. London Math. Soc.*, 17:75–115, 1918.
- [12] G. H. Hardy, Edward M. Wright, and Andrew Wiles. *An Introduction to the Theory of Numbers*. Oxford University Press, 2008.

- [13] Václav Havel. A remark on the existence of finite graphs. *Časopis pro pěstování matematiky*, 80(4):477–480, 1955.
- [14] Antal Iványi, Gergő Gombos, Loránd Lucz, and Tamás Matuszka. Parallel enumeration of degree sequences of simple graphs ii. *Acta Universitatis Sapientiae, Informatica*, 5(2):245–270, 2013.
- [15] Fredrik Johansson. Efficient implementation of the hardy-ramanujan-rademacher formula. *LMS Journal of Computation and Mathematics*, 15:341–359, 10 2012.
- [16] Axel Kohnert. Dominance order and graphical partitions. *Electronic Journal of Combinatorics*, 11(1), 2004.
- [17] Axel Kohnert. Number of different degree sequences of a graph with no isolated vertices. http://www.mathe2.uni-bayreuth.de/axel/magdeburg06_number_of_different_degree_sequences.pdf, 2016. Accessed: 2016-01-20.
- [18] N.V.R. Mahadev U.N. Peled. *Threshold Graphs and Related Topics*. North Holland, 1995.
- [19] Boris Pittel. Confirming two conjectures about the integer partitions. *Journal of Combinatorial Theory, Series A*, 88(1):123–135, 1999.
- [20] Boris Pittel. Asymptotic joint distribution of the extremities of a random young diagram and enumeration of graphical partitions, 2016.
- [21] G. Pólya. Kombinatorische anzahlbestimmungen für gruppen, graphen und chemische verbindungen. *Acta Mathematica*, 68(1):145–254, 1937.
- [22] J. Howard Redfield. The theory of group-reduced distributions. *American Journal of Mathematics*, 49(3):433–455, 1927.
- [23] Robert W. Robinson. Enumeration of non-separable graphs. *Journal of Combinatorial Theory*, 9(4):327–356, 1970.
- [24] Øystein J. Rødseth, James A. Sellers, and Helge Tverberg. Enumeration of the degree sequences of non-separable graphs and connected graphs. *Eur. J. Comb.*, 30(5):1309–1317, July 2009.
- [25] C. C. Rousseau and F. Ali. A note on graphical partitions. *Journal of Combinatorial Theory, Series B*, 64(2):314–318, 1995.
- [26] E. Ruch and I. Gutman. The branching extent of graphs. *Journal of Combinatorics, Information and System Sciences*, 4(4):285–295, 1979.
- [27] Frank Ruskey, Robert Cohen, Peter Eades, and Aaron Scott. Alley cats in search of good homes. In *25th S.E. Conference on Combinatorics, Graph Theory, and Computing*, volume 102, pages 97–110. Congressus Numerantium, 1994.

- [28] Gerard Sierksma and Han Hoogeveen. Seven criteria for integer sequences being graphic. *Journal of Graph Theory*, 15(2):223–231, 1991.
- [29] N. J. A. Sloane. The on-line encyclopedia of integer sequences. <https://oeis.org/A095268>, 2016. Accessed: 2016-01-18.
- [30] Richard Stanley. A zonotope associated with graphical degree sequences. In *Applied Geometry and Discrete Mathematics*, volume 4, pages 555–570. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1991.
- [31] D. L. Wang and D. J. Kleitman. On the existence of n -connected graphs with prescribed degrees ($n \geq 2$). *Networks*, 3(3):225–239, 1973.