

Graph realizations constrained by skeleton graphs

Erdős, Péter L.; Hartke, Stephen G.; van Iersel, Leo; Miklós, István

Publication date

2017

Document Version

Final published version

Published in

The Electronic Journal of Combinatorics

Citation (APA)

Erdős, P. L., Hartke, S. G., van Iersel, L., & Miklós, I. (2017). Graph realizations constrained by skeleton graphs. *The Electronic Journal of Combinatorics*, 24(2), 1-18. Article #P2.47.

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Graph realizations constrained by skeleton graphs*

Péter L. Erdős[†]

MTA Rényi Institute of Mathematics,
Reáltanoda u 13-15
Budapest, 1053 Hungary
erdos.peter@renyi.mta.hu

Stephen G. Hartke[‡]

Dept. Mathematical and Statistical Sciences,
University of Colorado Denver,
Colorado, Denver, USA
stephen.hartke@ucdenver.edu

Leo van Iersel[§]

Delft Institute of Applied Mathematics,
Delft University of Technology,
PO-box 5, 2600AA, Delft, Netherlands
l.j.j.v.iersel@gmail.com

István Miklós[†]

MTA Rényi Institute of Mathematics,
Reáltanoda u 13-15
Budapest, 1053 Hungary
miklos.istvan@renyi.mta.hu

Submitted: Jul 31, 2015; Accepted: Jun 19, 2017; Published: Jun 30, 2017

Mathematics Subject Classifications: 05C07

Abstract

In 2008 Amanatidis, Green and Mihail introduced the *Joint Degree Matrix* (JDM) model to capture the fundamental difference in *assortativity* of networks in nature studied by the physical and life sciences and social networks studied in the social sciences. In 2014 Czaparka proposed a direct generalization of the JDM model, the *Partition Adjacency Matrix* (PAM) model. In the PAM model the vertices have specified degrees, and the vertex set itself is partitioned into classes. For each pair of vertex classes the number of edges between the classes in a graph realization is prescribed. In this paper we apply the new *skeleton graph* model to describe the same information as the PAM model. Our model is more convenient for handling problems with low number of partition classes or with special topological restrictions among the classes. We investigate two particular cases in detail: (i) when there are only two vertex classes and (ii) when the skeleton graph contains at most one cycle.

Keywords: degree sequences; Joint Degree Matrix; Partition Adjacency Matrix; skeleton graph; forbidden edges; Tutte gadget; Edmonds's blossom algorithm

*This research started when SGH and LvI visited the MTA A. Rényi Institute of Mathematics, Budapest in the Fall of 2013.

[†]Partly by the Hungarian National Research, Development and Innovation Office NKFIH, under the grants K 116769 and SNN 116095.

[‡]Partly supported by a U.S. Fulbright Scholar Fellowship and by a grant from the Simons Foundation (#316262 to Stephen Hartke).

[§]Partly funded by the Netherlands Organisation for Scientific Research (NWO), including Veni grant 639.071.106 and Vidi grant 639.072.602 and by the 4TU Applied Mathematics Institute.

1 Introduction

In the last fifteen years, the exponential development of network theory has raised the *practical* problem of realizing and sampling large graphs with given degree sequences. Finding a realization of a given degree sequence (among simple graphs or graphs with a given maximum number of parallel edges or/and loops) has long been shown to be an easy problem. Havel [12] first solved the problem in 1957, and his algorithm was reinvented by Hakimi [11] in 1962. An even older but less efficient way to find realizations can be derived from Tutte's f -factor results [25, 26]. Another method was due to Paul Erdős and Gallai [7] in 1960, but the resulting algorithm was derived from Havel's approach. All these methods lack the ability of generating all (or even a large number of different) realizations. The problem of determining if there exists a graph with given degree sequence and satisfying other specified conditions will be called in this paper the **realization** problem.

In many situations, it is not feasible to generate all realizations, as the number of possible realizations can be exponential or larger in the length of the degree sequence. In this case, practical applications may require reasonable sampling methods of the "typical" realizations. A common approach is to use Markov Chain Monte Carlo methods, which require some simple operations that transform one realization into another, slightly different realization. Additionally, it must be possible to transform any given realization into any other using these operations. This particular subproblem of the sampling procedure will be called in this paper the **connectivity** problem (also known as the irreducibility problem in the context of Markov Chain Monte Carlo processes for sampling random realizations).

In modern graph theory the first such manipulation was Havel's *swap* operation from [12]. (The terminology and notation used in this paper will be introduced in detail in Section 2.) It is interesting to remark that his method was applied already by Petersen [18] in 1891, who essentially showed that any realization of a given degree sequence can be transformed into any other realization of the same degree sequence by a series of such swap operations. In 1951 Senior [20] also used this approach to construct possible hydrocarbon molecules with given atomic composition. For bipartite graphs it was done by Ryser [19] in 1963, and all these results have been invented again and again.

Recently a large number of real-world social and biological networks were studied in detail. One important distinction between these two types of networks lies in their overall structure: social networks typically have a few very high degree vertices and many low degree vertices with high *assortativity* (where a vertex is likely to be adjacent to vertices of similar degree), while biological networks are generally *disassortative* (in which low degree vertices tend to attach to those of high degree). It is easy to see that the *degree sequence* alone cannot capture these differences. There are several approaches to address this problem. See the paper of Stanton and Pinar [22] for a detailed description of the current state-of-the-art.

One way to address this problem is the **joint degree matrix** model (or JDM for short). This model is more restrictive than the degree distribution, but it provides a

way to enhance results based on degree distribution. It was introduced by Amanatidis, Green and Mihail [1] and Stanton and Pinar [22]. In essence, the JDM specifies the exact *number* of edges between vertices of degrees i and j . More precisely, a *joint degree matrix* $\mathcal{J}(G) = [\mathcal{J}_{ij}]$ of the graph $G = (V, E)$ is a $\Delta \times \Delta$ matrix (Δ is the maximum degree of G) where $\mathcal{J}_{ij} = |\{xy \in E(G) : d(x) = i \text{ and } d(y) = j\}|$. It is clear that the degree sequence of the graph is determined by its JDM:

$$(\text{the number of vertices of degree } i) = \frac{1}{i} \left(\mathcal{J}_{ii} + \sum_{\ell=1}^{\Delta} \mathcal{J}_{i\ell} \right). \quad (1)$$

The novelty of this definition is that values \mathcal{J}_{ij} are exact numbers, and not expectations, like in earlier approaches, see for example [16].

The existence problem for the JDM model is not hard: already Patrinos and Hakimi [17] presented in 1976 an Erdős-Gallai-type theorem for joint degree matrices, essentially characterizing precisely those matrices which are the joint degree matrix for some graph, though using different terminology. Another proof for this result was given in [1], see also [2]. Stanton and Pinar [22] gave a separate, constructive proof for this theorem, which builds a particular graph that has a given matrix as its JDM. Czaparka, Dutle, Erdős and Miklós [3] presented a simpler proof using a construction algorithm that can create *every* graph with a given JDM. See also [10].

The connectivity problem for the JDM model proved to be more complicated. Stanton and Pinar [22] solved it for the space of all multigraph realizations. For simple graphs it was resolved affirmatively by Czaparka, Dutle, Erdős and Miklós [3].

The JDM model suggests a more general restricted degree sequence problem: the **partition adjacency matrix** model (or PAM for short). In this generality, it was introduced by É. Czaparka [4]. Let $\Pi = (P_1, \dots, P_k)$ be a partition of the vertex set V of the graph G . Let \mathbf{d} be the degree sequence of G , and let M be the following $k \times k$ matrix: if $i \neq j$ then the entry $M_{i,j}$ is the number of edges in the bipartite subgraph $G[P_i, P_j]$, while $M_{i,i}$ is the number of edges within the induced subgraph $G[P_i]$. The matrix M is called the *partition adjacency matrix* of the graph G for the partition Π . Clearly the *PAM-problem* is: we are given a positive integer sequence on the partitioned V and a matrix M and we want to decide whether there is a graph G with the given degree sequence and with the given PAM.

The joint degree matrix determines the degree sequence itself by equation (1). Therefore the JDM is clearly an instance of the PAM-problem. The PAM problem in full generality is probably quite complicated: when we have, say, $\sqrt{|V|}$ partition classes, then the problem is conjectured to be NP-complete.

In this paper we will consider an auxiliary structure to describe the PAM problem. This is a more convenient description when there are only a small, say, linear number of items in M which are not zero. This object also provides structural properties of the edges among the vertex partitions. This description is based on the notion of a *skeleton graph*, and it is described in detail in Section 2.

We study two particular skeleton graphs in detail: the first one consists of two partition classes with edges inside the classes allowed, while in the second one each of its components contains at most one cycle while the classes have no edges inside. In both cases we show how to quickly construct graphical realizations in all feasible cases. We also consider whether the space of all realizations are connected. The answers in both cases are almost affirmative: the space is connected if we use swaps as well as an additional operation called *double swaps*.

2 Definitions and tools

Let $G = (V, E)$ be a *simple* graph with vertex set $V = \{v_1, \dots, v_n\}$ and edge set E (no multiple edges or loops). Denote $\mathbf{d}(G) = (d(v_1), \dots, d(v_n))$ its *degree sequence*. This sequence is not ordered in any way. The graph G is called a *realization* of the previous sequence. A sequence $\mathbf{d} = (d_1, \dots, d_n)$ of nonnegative integers is **graphical** if it has at least one realization. The set of all realizations of a graphical degree sequence \mathbf{d} is denoted by $\mathbb{G}(\mathbf{d})$.

We consider realizations of a degree sequence $\mathbf{d} = (d_1, \dots, d_n)$ that are restricted by a “skeleton” graph \mathcal{S} in the following way. We fix a partition $\Pi = \{U, W, \dots\}$ of V .

A **skeleton graph** is an edge-weighted graph $\mathcal{S} = (\Pi; \mathcal{B}, w)$ on the partition classes in Π with possible loops. We will refer to a vertex of \mathcal{S} as a **class** and the edges of the skeleton graph \mathcal{S} are referred to as **bones**. We will use the following, very natural, notation: for $U, W \in \Pi$, if the pair UW is a bone in \mathcal{B} , then $w(UW)$ is its weight (in \mathcal{S}). If $U = W$ then it is a loop. If UW is not a bone in \mathcal{B} then $w(UW) = 0$. Furthermore, $G[U, W]$ is the induced bipartite subgraph in G while $G[U, U] = G[U]$ is the induced simple graph within U . The graphs $G[U, W]$ are the *component graphs* of G . When the partition classes in Π consist of all vertices with the same degrees, then all the introduced notions are equivalent to the notions used for the Joint Degree Matrix model in [3].

We will say that the realization G of degree sequence \mathbf{d} is **consistent** with the skeleton graph \mathcal{S} if

$$\forall U, W \in \Pi \quad w(UW) = |E(G[U, W])|. \quad (2)$$

We say that the pair $(\mathbf{d}, \mathcal{S})$ is **graphical** if there exists a realization G of \mathbf{d} which is consistent with \mathcal{S} . See for example in Figure 1 a skeleton graph \mathcal{S} and a realization G that is consistent with \mathcal{S} . We will consider several different realizations of a degree sequence, even ones that are not consistent with the skeleton graph. However we will not consider realizations which have edges in a component graph without a corresponding bone. Therefore, we call a pair of vertices within a component graph with a corresponding bone a *chord*, and each other pair of vertices a *non-chord*. So a chord is a pair of vertices which may form an edge in a consistent realization. A realization G of a degree sequence \mathbf{d} is **weakly consistent** with \mathcal{S} if all of the edges of G are chords. Here property (2) may not hold. Although the existence problem of weakly consistent realizations is not interesting in itself, we will use it as a tool. Before we discuss this, we recall some details about Tutte’s f -factor theorem and its applications.

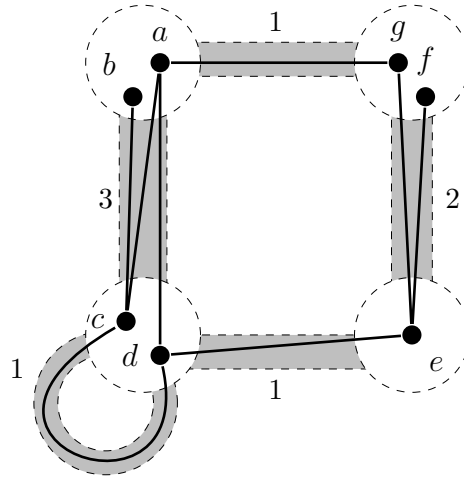


Figure 1 The dashed circles illustrate the classes of a skeleton graph \mathcal{S} , while its bones are indicated by gray dashed tubes. The dots and solid lines show a graph G that is consistent with \mathcal{S} .

In 1947 Tutte completely characterized the graphs with perfect matchings (see [24]). In 1952 he generalized this result for the so called f -factor problem. Two years later, in 1954, Tutte found the following brilliant way to reduce the problem of finding an f -factor in a given graph to finding a perfect matching (a 1-factor) in an auxiliary graph ([26]). Let G be a simple graph and let $f(v)$ be a non-negative integer for each $v \in V$. Then a subgraph F of G in which each vertex $v \in V$ has degree $f(v)$ is an f -factor of G . Any f -factor in G can be represented as a perfect matching in the auxiliary graph $\mathbb{T}(G, f)$:

$$V(\mathbb{T}) = \{v_1, \dots, v_{d(v)-f(v)} \mid v \in V(G)\} \cup \{e_v, e_u \mid vu = e \in E(G)\} \quad (3)$$

$$E(\mathbb{T}) = \{v_i e_v \mid i = 1, \dots, d(v) - f(v); e = vu \in E(G)\} \cup \{e_v e_u \mid e = vu \in E(G)\}. \quad (4)$$

It is easy to see that there is a natural bijection between the f -factors in G on one hand and the perfect matchings in $\mathbb{T}(G, f)$ on the other hand. More precisely, given the perfect matching M in $\mathbb{T}(G, f)$, the requested subgraph in G is

$$\{e \in E \mid e = vu, e_v e_u \in M\}.$$

In 1965, Edmonds described an effective algorithm to find a maximum matching in G ([5]). Then, in the same year, he extended his approach for edge-weighted graphs ([6]): his blossom algorithm finds a maximum-weight perfect matching in strongly polynomial time. We will use this result extensively in this paper. The classical existence problem for degree sequences can be easily solved by Tutte's f -factor theorem and Edmonds' algorithm: take as graph G the complete graph on n vertices and for the function f the

degree sequence. (See for example [13] for an outstanding application.) However, this method is less efficient than the methods based on Havel’s observation, and it cannot be used to find all possible realizations (see for example [14]). Nevertheless, Edmonds’ blossom algorithm is excellent to find weakly consistent realization to any skeleton graph:

Observation 2.1. *For any degree sequence \mathbf{d} and any skeleton graph $\mathcal{S} = (\Pi; \mathcal{B}, w)$, one can decide in strongly polynomial time whether there exists a realization of \mathbf{d} that is weakly consistent with \mathcal{S} .*

Proof. The graph G in the f -factor problem consists of all chords defined by the skeleton graph, while the f -function is equal to the given degree sequence. \square

It is clear that the f -factor approach cannot directly find consistent realizations for any “reasonable” skeleton graph problem: it has no control over the exact number of edges in the component graphs. We even cannot enforce that all bones contain at least one edge in the derived realization. We need additional ideas to find consistent realizations. For that end we will extensively use some restricted versions of Havel’s swap operation.

Definition 2.2 (unrestricted / restricted / \mathcal{S} -preserving swap operations).

- (R1) Let G be a realization of the graphical sequence \mathbf{d} , if a, b, c and d are vertices of G satisfying $ab, cd \in E$ and $bc, ad \notin E$, then the graph $G' = (V, E')$ with $E' = E \cup \{bc, ad\} \setminus \{ab, cd\}$ is another realization of \mathbf{d} . This **swap** operation, denoted by $ab, cd \Rightarrow bc, ad$, was introduced by Havel [12]. It is also known, for example, as *switch* or *rewiring* or *infusion* operation.
- (R2) Let \mathcal{S} be a skeleton graph, and let G be a realization of \mathbf{d} that is weakly consistent with \mathcal{S} . If all vertex pairs in (R1) are chords, then G' will be weakly consistent with \mathcal{S} . Then the operation is a **restricted swap** operation.
- (R3) If G is consistent with the skeleton graph \mathcal{S} and G' is also consistent with \mathcal{S} , then this operation is an **\mathcal{S} -preserving swap** operation.

As we mentioned earlier already Petersen proved ([18]) that any realization of a given degree sequence can be transformed into another one by consecutive unrestricted swap operations. Havel’s result gives a rather crude algorithm to find such a swap sequence: the number of steps may be twice the number of edges in the worst case. It is very natural to ask what the minimum length of such a swap sequence is. This question was studied in details by Erdős, Király and Miklós ([8]). Next we will summarize the main findings of their paper:

Regular swap sequences: Let G and G' be two realizations of the degree sequence \mathbf{d} . The symmetric difference $\nabla = E(G) \Delta E(G')$ of their edges has a natural 2-coloration: an edge in ∇ is **red** or **blue** depending on whether it belongs to G or G' . Denote by $r(G, G')$ the number of red edges in the symmetric difference (which is of course also the number of blue edges).

The following statement is a simple observation:

Lemma 2.3. *Every vertex in ∇ has an equal number of red and blue adjacent edges. Moreover, the symmetric difference ∇ can be decomposed into even length alternating (with respect to the coloration) circuits (closed walks), where no circuit contains any vertex more than twice.* \square

Consider now two realizations G and G' such that ∇ is one alternating circuit, C . The next lemma is one of the central results in [8]:

Lemma 2.4. *There exists a sequence of consecutive swap operations transforming realization G into G' with the following properties: (i) Along the process every swap is applied for vertex pairs belonging completely to $V(C)$. (ii) If G_1 and G_2 are two consecutive realizations along the sequence then $r(G_1, G') - r(G_2, G') \in \{0, 1, 2\}$. (iii) The length of this swap sequence is $r(G, G') - 1$.* \square

The described swap sequence is called a **regular swap sequence**.

Theorem 2.5 (Erdős–Király–Miklós, [8]). *Let G and G' be arbitrary realizations of degree sequence \mathbf{d} . Every shortest possible swap sequence can be reordered such that this realigned sequence is identical with a series of subsequent regular swap sequences, corresponding to a circuit decomposition of the symmetric difference ∇ . The length of this swap sequence is*

$$r(G, G') - \text{the maximum possible number of circuits in a decomposition,}$$

and hence is at most $r(G, G') - 1$.

With some lack of precision we also call the swap sequence described above as a **regular swap sequence**, and a shortest regular swap sequence, respectively. Two further useful observations:

Remark 2.6.

- (i) Any particular alternating circuit in ∇ can be extended into a complete decomposition of ∇ .
- (ii) If an arbitrary swap sequence transforms G into G' then the *inverse* swap sequence transforms G' into G (here we do not define the notion inverse, because it is self-evident).

The following is easily verified.

Observation 2.7. *In an \mathcal{S} -consistent realization G , a swap $ab, cd \Rightarrow bc, ad$ is \mathcal{S} -preserving if and only if either a and c , or b and d are in the same class of \mathcal{S} .*

One of our motivating questions is whether the space of realizations of a graphical sequence consistent with a given skeleton graph is connected using swaps. In the classic cases this is true (see, for example, Theorem 2.5) as well as in the Joint Degree Matrix case (see [3]). However, as we will see later on, there is not so neat answer for the skeleton graph problems. The obvious reason for this is that Theorem 2.5 does not apply for this case, since a regular swap sequence does not necessarily use \mathcal{S} -preserving swap operations.

3 Graph realizations with a given number of edges crossing a given bipartition

We start our investigations with one of the most simple skeleton graphs. Let $V = \{v_1, \dots, v_n\}$ be a vertex set, \mathbf{d} a degree sequence, $\Pi^2 = (U, W)$ a partition of V , $k \in \mathbb{N}$ and let $\mathcal{S}(k) = (\Pi^2; \mathcal{B}, w)$ be a skeleton graph with two vertices, a weight- k UW bone and two loops, whose weights are completely defined by \mathbf{d} and k . The edges with end vertices in both classes are called the **crossing** edges.

The conventional description is the following: Given the degree sequence \mathbf{d} , a bipartition of the vertex set and a natural number k , decide if there exists a graph that realizes the given degree sequence and has precisely k crossing edges.

When k is equal to the total number of edges, i.e. both classes induce the empty graph, then this coincides with the usual bipartite degree sequence problem.

When W has no inner edges, then no direct greedy method (like Havel's lemma) is known to construct a consistent realization. The reason for that is simple: we just do not know the U -side of the bipartite degree sequence in the component graph $G[U, W]$. However, the Tutte - Edmonds method provides an effective solution for the existence problem.

3.1 Existence

It is clear that any realization of \mathbf{d} is automatically weakly consistent. In this subsection we consider the existence problem for consistent realizations.

Theorem 3.1. *We can decide in polynomial time whether there exists a realization of \mathbf{d} that is consistent with $\mathcal{S}(k)$.*

The number of crossing edges in a realization G is denoted $\epsilon(G)$. The set of all realizations of degree sequence \mathbf{d} with ℓ crossing edges is denoted $\mathbb{G}_\ell(\mathbf{d})$.

Let $G \in \mathbb{G}_\ell(\mathbf{d})$ and let G' be the realization derived from G by the swap operation $ac, bd \Rightarrow bc, ad$. By simple case analysis it is easy to see that

$$|\epsilon(G) - \epsilon(G')| \in \{0, 2\}. \quad (5)$$

Lemma 3.2. *Let G and G' be realizations of \mathbf{d} with $\epsilon(G) < \epsilon(G')$. Then*

- (i) $\epsilon(G) \equiv \epsilon(G') \pmod{2}$;
- (ii) for all $\ell \in \{\epsilon(G), \epsilon(G) + 2, \dots, \epsilon(G')\}$, there exists a realization $G'' \in \mathbb{G}_\ell(\mathbf{d})$.

Proof. By Theorem 2.5 there is a regular swap sequence turning G into G' . By equation (5), the realizations in this sequence either all have an even number of crossing edges or all have an odd number of crossing edges. Moreover, this sequence hits a realization G'' with $\epsilon(G'') = \ell$ for any $\epsilon(G) \leq \ell \leq \epsilon(G')$ with $\ell = \epsilon(G) \pmod{2}$. \square

Denote by $\epsilon_m(\mathbf{d})$ the *minimum* value of ℓ such that \mathbb{G}_ℓ is not empty, and similarly denote by $\epsilon_M(\mathbf{d})$ the *maximum* value. By Lemma 3.2 the sets

$$\mathbb{G}_{\epsilon_m(\mathbf{d})}(\mathbf{d}), \mathbb{G}_{\epsilon_m(\mathbf{d})+2}(\mathbf{d}), \dots, \mathbb{G}_{\epsilon_M(\mathbf{d})}(\mathbf{d}) \quad (6)$$

of all weakly consistent realizations are not empty while all other sets $\mathbb{G}_\ell(\mathbf{d})$ are empty.

Edmonds' blossom algorithm [6] applied to the Tutte gadget $\mathbb{T}(K_n, \mathbf{d})$ can easily find the minimum and maximum values $\epsilon_m(\mathbf{d})$ and $\epsilon_M(\mathbf{d})$ together with the corresponding realizations consistent with $\mathcal{S}(\epsilon_m(\mathbf{d}))$ and $\mathcal{S}(\epsilon_M(\mathbf{d}))$, respectively.

We first assign weight 0 to all crossing edges in $\mathbb{T}(K_n, \mathbf{d})$, while to all other edges we assign weight 1. Then any maximum weight perfect matching corresponds to a realization G which contains the minimum possible number of crossing edges. Therefore $\epsilon_m(\mathbf{d}) = \epsilon(G)$. To determine ϵ_M we just reverse the edge weights. So the solution of the maximum weight matching problem for this graph provides the value $\epsilon_M(\mathbf{d})$ and a realization $G \in \mathbb{G}_{\epsilon_M(\mathbf{d})}$.

The formula (6) gives all other possible ϵ values while the proof of Lemma 3.2 (ii) describes the way to find realizations with specific ϵ -values. \square

This proof seems to be easy and straightforward. However, the situation is more complicated. First of all, as far as the authors are aware, this is the very first solved degree sequence type problem without some direct, greedy type solution. Secondly, it is somewhat unusual to solve an existence problem, where the two extremal solutions can be found directly, while the solutions inbetween can be inferred only indirectly from the extremal solutions. Finally, we think that it is a small miracle that this proof works at all. Consider the following, slightly different question: the vertices, equipped by a degree sequence \mathbf{d} , are partitioned into three classes: U, W, Z . We are looking for a realization G of \mathbf{d} which has exactly k edges between U and W . There is no any other restriction. (Of course this problem does not belong to the skeleton graph problem class, but the difference is tiny.) For this problem, Lemma 3.2 does not hold. Therefore, finding realizations with minimum m and maximum M number of crossing edges does not help to solve the problem if $m < k < M$. Actually, this problem seems to be quite hard, and the authors were not able to provide a polynomial-time algorithm solving it despite of serious efforts.

3.2 Connectivity

Now consider two realizations G, G' of degree sequence \mathbf{d} , consistent with the skeleton graph $\mathcal{S}(k) = (\Pi^2; \mathcal{B}, w)$. Is it true that in such a situation there always exists a sequence of \mathcal{S} -preserving swaps that turn G into G' ? The following counter example shows that the answer to the above question is "no".

Theorem 3.3. *There exist graphs G, G' that have the same vertex set V , the same degree sequence \mathbf{d} and that are both consistent with the skeleton graph $\mathcal{S}(k) = (\Pi^2; \mathcal{B}, w)$ described above, such that there exists no sequence of \mathcal{S} -preserving swaps that turn G into G' .*

Proof. We construct an example as follows. Let $\mathbf{d} = (6, 6, 3, 3, 3, 3, 1, 1)$ be the degrees of vertices $(u_3, w_7, u_2, w_6, u_1, w_5, u_0, w_4)$. Let \mathcal{S} be the skeleton graph with $\Pi = \{U, W\}$

with $U = \{u_0, u_1, u_2, u_3\}, W = \{w_4, w_5, w_6, w_7\}$ and with bone UW with weight $k = 7$. Then both vertex classes must contain 3 edges. In Figure 2 we show all realizations of this degree sequence that are consistent with \mathcal{S} . There is a single \mathcal{S} -preserving swap that turns G_1 into G_2 , namely the swap $u_2w_5, u_1w_6 \Rightarrow u_1w_5, u_2w_6$. However, there is no \mathcal{S} -preserving swap that turns G_1 into G_3 or G_2 into G_3 . (See Figure 3 where the red and blue chords can be found in only one realization, while the black chords are in both.) Hence, there is no sequence of \mathcal{S} -preserving swaps that turns G_1 or G_2 into G_3 . \square

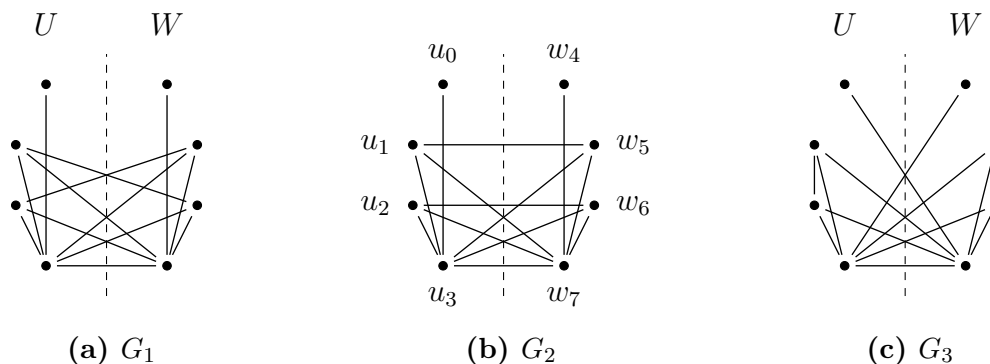


Figure 2 There is no sequence of \mathcal{S} -preserving swaps that turns G_1 or G_2 into G_3 .

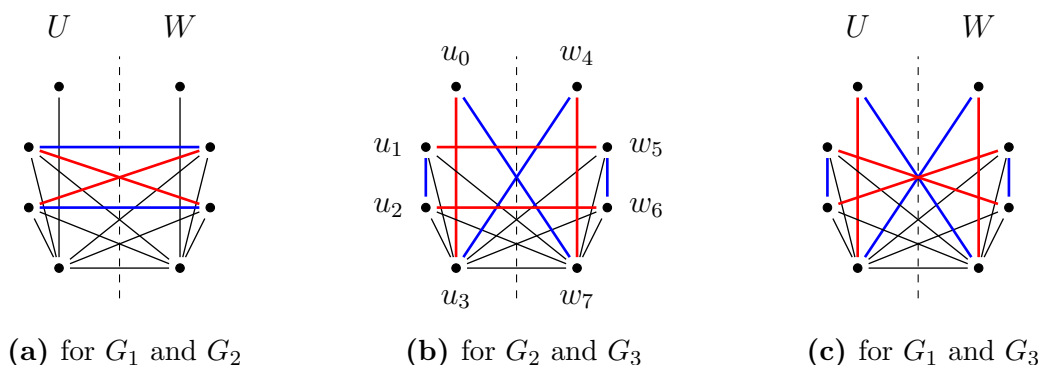


Figure 3 The red and blue chords come from one realization, while the black ones come from both. The realizations themselves come from Figure 2.

Motivated by Theorem 3.3, we define a **double swap** as the simultaneous application of two disjoint swaps, i.e., if a, b, c, d, e, f, g, h are eight distinct vertices of graph $G = (V, E)$ such that $ab, cd, ef, gh \in E$ and $bc, ad, fg, eh \notin E$, then the result of applying the double swap $ab, cd, ef, gh \Rightarrow bc, ad, fg, eh$ is the graph $G' = (V, E \cup \{ab, cd, ef, gh\} \setminus \{bc, ad, fg, eh\})$. A double swap is \mathcal{S} -preserving, for some skeleton graph \mathcal{S} , if G' is consistent with \mathcal{S} .

Theorem 3.4. *For every two graphs G, G' that have the same vertex set V , the same degree sequence \mathbf{d} and that are both consistent with the same skeleton graph $\mathcal{S}(k) =$*

$(\Pi^2; \mathcal{B}, w)$, there exists a sequence of \mathcal{S} -preserving swaps and double swaps that turn G into G' .

Proof of Theorem 3.4. The proof is by induction on $r(G, G')$. Let's recall that this is the number of red edges in $G \Delta G'$ which is $\frac{1}{2}|E(G) \Delta E(G')|$. If $r(G, G') = 0$ then $G = G'$ and we are done. Moreover, if $r(G, G') = 2$ then we can turn G into G' by applying the single swap consisting of the red and blue edges of ∇ and we are again done. (It is easy to see that $r(G, G')$ cannot be 1.) Hence, assume $r(G, G') \geq 3$. In a sequence of lemmas we will show that we can always find an \mathcal{S} -preserving swap and/or double swap sequence which reduces $r(G, G')$. In each lemma we identify an alternating circuit in ∇ s.t. the regular swap subsequence processing this circuit will apply only \mathcal{S} -preserving swaps, and will hence satisfy the conditions described in Theorem 3.4.

Lemma 3.5. *If one can find an alternating circuit $C \subset \nabla$ completely within a vertex class, say, U , then the regular swap subsequence processing C (see Theorem 2.5) provides \mathcal{S} -preserving swaps at each step.*

Proof. Indeed, the regular swap sequence can start with that particular circuit—see Lemma 2.6(i)—and the process uses chords of this circuit, therefore all swap operations happen inside class U and hence all swaps are \mathcal{S} -preserving. \square

From now on we assume that there exists no alternating circuit completely within any vertex class. To make the references easier for different type of chords in the realizations, we introduce two further notions: we say that a chord is **black** if it is an edge in both G and G' . It is **white** if it is an edge neither in G nor in G' .

Lemma 3.6. *Assume that there exists a red crossing edge and a blue crossing edge that share an endpoint. Then we can identify an alternating circuit C such that the regular swap subsequence which processes C provides only \mathcal{S} -preserving swaps.*

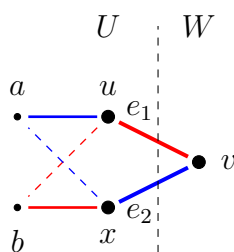


Figure 4 A red crossing chord and a blue crossing chord that share an endpoint. The red chord is an edge in G but not in G' . Analogously the blue chord is an edge in G' but not in G . Either au is blue but ax is not blue, or bx is red but bu is not red.

Proof. Suppose that $e_1 = uv$ is a red crossing edge and $e_2 = xv$ is blue crossing edge, where $u, x \in U$ and $v \in W$; see Figure 4. We first show that there is a vertex $a \in V(G) \setminus \{u, v, x\}$ such that au is blue but ax is not blue, or there is a vertex $b \in V(G) \setminus \{u, v, x\}$ such that

bx is red but bu is not red. If no such vertex a exists, then $d_b(u) < d_b(x)$, and if no such vertex b exists, then $d_r(x) < d_r(u)$. But since $d_r(u) = d_b(u)$ and $d_r(x) = d_b(x)$ we have $d_r(u) = d_b(u) < d_b(x) = d_r(x)$, which contradicts the previous sentence.

Hence such vertex a or vertex b exists. Without loss of generality assume that there exists a vertex a such that au is blue but ax is not blue.

If ax is red or black, then we apply the \mathcal{S} -preserving swap au, uv, xv, ax to G to obtain G^* , a consistent realization. Since $r(G^*, G') < r(G, G')$ therefore the induction applies.

If ax is white, then we apply the swap $au, xv \Rightarrow ax, uv$ for realization G . (Here we apply Lemma 2.6 (ii).) For the derived realization G^* the number of red edges within $E(G) \Delta E(G^*)$ is smaller than $r(G, G')$ and induction applies. \square

From now on we assume that there exist no adjacent red and blue crossing edges.

Lemma 3.7. *Assume there exists a red-blue alternating trail T where the first chord of T is a red crossing one, the last chord is a blue crossing one, and the other chords of T are not crossing ones. (Then T consists of at least four chords.) Then we can identify an \mathcal{S} consistent realization G^* with the property, that $|E(G) \Delta E(G^*)| < 2r(G, G')$ and $|E(G^*) \Delta E(G')| < 2r(G, G')$, therefore the inductive hypothesis applies.*

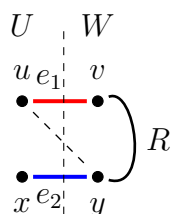


Figure 5 A red crossing chord and a blue crossing chord connected by a red-blue alternating chord trail in G . Recall, that a red chord is an actual edge in G but a non-edge in G' and a blue chord is a non-edge in G but an edge in G' .

Proof. Suppose that such an alternating trail T exists. Let $e_1 = uv$ be the red crossing edge in T and $e_2 = xy$ be the blue crossing edge, where $u, x \in U$, $v, y \in W$, and the other edges of T lie within W . Let R be the subtrail of T from v to y formed from T by removing e_1 and e_2 , then R contains at least two chords. See Figure 5 for a picture.

Consider the pair uy . Since Lemma 3.6 does not apply, uy is a black or white chord. If uy is black, then denote by C the chord circuit (R, vu, uy) . This has exactly two crossing chords, one is red (an edge in G but non-edge in G'), the other is blue (a non-edge in G but an edge in G'), and it alternates between red and blue chords: every second chord is red, the others are blue. Denote by C^* the chord circuit derived from C by switching blue and red chords along C and denote by G^* the graph derived from G' by exchanging C and C^* . Then G^* is clearly another realization of \mathbf{d} and since C^* has two crossing chords, one is an edge, the other is a non-edge, therefore, G^* is consistent with the skeleton graph $\mathcal{S}(k)$. Since trail R contains at least one red chord, the number of red edges in $E(G) \Delta E(G^*)$ is

smaller than $r(G, G')$ and, consequently, the inductive assumption applies for G and G^* : there exists a sequence Σ_1 of \mathcal{S} -preserving swaps and double swaps transforming G into G^* . Furthermore, since T was not closed, the number of red edges in $E(G') \Delta E(G^*)$ is smaller than $r(G, G')$ and, therefore, the inductive assumption applies for G^* and G' as well: there exists a sequence Σ_2 of \mathcal{S} -preserving swaps and double swaps transforming G^* into G' . So the swap-sequence $\Sigma_1 \circ \Sigma_2$ transforms G into G' via \mathcal{S} -preserving operations.

If uy is white, then we consider exactly the same circuit C and let C^* be derived in the same way. However, G^* is now derived from G by exchanging C and C^* . Again G^* is consistent with the skeleton graph $\mathcal{S}(k)$. Since $r(G, G^*) = |T|/2 < r(G, G')$, there exists an \mathcal{S} -preserving swap sequence Σ_1 from G to G^* . Similarly there exists an \mathcal{S} -preserving swap sequence Σ_2 transforming G^* into G' . So the swap-sequence $\Sigma_1 \circ \Sigma_2$ transforms G into G' via \mathcal{S} -preserving operations. \square

Now we are ready to finish the proof of Theorem 3.4: Consider the symmetric difference ∇ . Assume at first there is a connected component in ∇ containing both red and blue crossing chords. Then they are connected with an alternating chord trail and traversing it will reveal a red and blue crossing chord pair as in Figure 5. Therefore we may assume that no connected component has both red and blue crossing chords. Next we decompose it into alternating chord circuits and let C_1 and C_2 be two of them, the first has red crossing chords while the second one has blue crossing chords. By our assumptions these two circuits are vertex disjoint.

Let Σ_1 denote a regular swap sequence processing C_1 and let Σ_2 denote the analogous regular sequence for C_2 . Now we execute Σ_1 step by step while the required swaps are \mathcal{S} -preserving. Let $\Sigma_1(i)$ denote the last performed operation. If the swap $\Sigma_1(j)$, where $j \leq i$, produced realization H and $r(H, G')$ is smaller than $r(G, G')$, then the inductive hypothesis applies. So we suppose here that the operations $\Sigma_1(1), \dots, \Sigma_1(i)$ only split the alternating chord circuits into smaller ones.

At this point we execute the swap sequence Σ_2 step by step while the required swaps are $\mathcal{S}(k)$ -preserving. Let $\Sigma_2(j)$ denote the last performed operation. We let H denote the current realization, which is consistent with $\mathcal{S}(k)$ and has $r(H, G')$ equal to $r(G, G')$.

Now we execute the **double swap** $\Sigma_1(i + 1)$ and $\Sigma_2(j + 1)$ which together produce a new realization H' which is consistent with $\mathcal{S}(k)$ and for which $r(H', G')$ is smaller than $r(G, G')$. The inductive hypothesis applies. This completes the proof of Theorem 3.4. \square

4 Multipartite graph realizations

This section considers skeleton graphs with more than two classes but without loops. First, we consider skeleton graphs that contain exactly one odd cycle.

Lemma 4.1. *Let V be the underlying vertex set with partition $\Pi = (U_1, \dots, U_{2k-1})$, let \mathbf{d} be a sequence of $|V|$ integers, and $\mathcal{S} = (\Pi; \mathcal{B}, w)$ a skeleton graph consisting of exactly one odd cycle $U_1U_2, U_2U_3, \dots, U_{2k-1}U_1$ with an otherwise undefined weight function w . Then*

there exists at most one weight function w for which $\mathbb{G}(\mathcal{S}(w))$ is not empty. (Here $\mathcal{S}(w)$ is a shorthand for $\mathcal{S} = (\Pi; \mathcal{B}, w)$.)

Proof. Assume that G is a realization of \mathbf{d} which is weakly consistent with skeleton graph $\mathcal{S} = (\Pi; \mathcal{B})$. (Recall: there is no edge in G outside \mathcal{B} .) Let $D(U)$ (for $U \in \Pi$) denote $D(U) = \sum_{u \in U} d(u)$. (This is the total degree of U in the skeleton graph.) Furthermore let α denote the number of edges in the bone U_1U_2 . Then there are $D(U_1) - \alpha$ edges in the bone U_1U_{2k-1} .

We can calculate the number of edges in the bone U_2U_3 : it is $D(U_2) - \alpha$. The number of edges along the bone U_3U_4 is $D(U_3) - D(U_2) + \alpha$. And so on: we can calculate the number of edges in all bones, one by one. We finish it to calculate the number of edges in the bone U_{2k-2}, U_{2k-1} which is some β , a linear function of α with coefficient 1. Finally we know that $\alpha + \beta = D(U_{2k-1})$. So α is fully determined. If a weakly consistent realization exists then the solution for α must provide non-negative integer values $w^*(U_iU_{i+1})$ for all bones. \square

The easy consequence is that all weakly consistent realizations are consistent with exactly the same weight function: each belongs to $\mathcal{S}(w^*)$.

By the above lemma, we can solve the existence problem for this particular skeleton graph by deciding if there exists a realization of \mathbf{d} that is weakly consistent with \mathcal{S} . But Observation 2.1 does exactly this for us.

Corollary 4.2. *There exists a polynomial algorithm to decide the existence of a consistent realization to the skeleton graph problem above.*

Before we proceed to even cycles we need some definitions and a result about restricted degree sequence problems from [9]. Let F be a subset of the vertex pairs from V . The other vertex pairs on V are called **chords**. Let \mathbf{d} be a degree sequence on V . We are interested in those realizations of \mathbf{d} which completely miss F (the *forbidden* set of non-chords). The set of all such realizations is denoted by $\mathbb{G}^F(\mathbf{d})$.

Consider a realization $G = (V, E) \in \mathbb{G}^F(\mathbf{d})$ where v_1, \dots, v_{2k} is an alternating (edge, non-edge, ... etc.) circuit of chords. Assume that all pairs $v_i v_j$ which would divide C into two even chord circuits (these are the pairs $i, j \in \{1, \dots, 2k\}$ with $j = i + 1 \pmod{2}$ and $|i - j| > 1$) are forbidden (they are not chords). Then the operation which exchanges the edges and non-edges along the circuit C is called an **F-swap**. It is clear that if $F = \emptyset$ then this notion coincides with Havel's swap notion. The following result follows directly from Theorem 2.3 of [9]: the space $\mathbb{G}^F(\mathbf{d})$ is connected under F -swaps. More precisely:

Theorem 4.3 ([9]). *Let $G, G' \in \mathbb{G}^F(\mathbf{d})$ be two realizations. Then there exists a sequence of F -swaps which turns G into G' . Moreover if the symmetric difference between G and G' is one alternating chord circuit, then all the F -swaps happen within the vertex set of C .* \square

We continue our investigations with considering skeleton graphs consisting of one even cycle: we use similar notations as before except that the last vertex partition is denoted by U_{2k} and the cycle is modified accordingly.

The existence problem for such skeleton graphs was originally raised for the case $k = 2$ by László A. Székely [23]; he also suggested a solution for this particular question.

Now we can do the same calculation here that happened in the proof of Lemma 4.1. However, the final equation contains no α , it is just an alternating sum of $D(U_i)$ s, an identity. Therefore there may exist several feasible values for α , and it is possible to find all feasible weight functions (a constant number) and to find at least one consistent realization for each feasible weight function in polynomial time. More precisely:

Lemma 4.4. *Let $\mathcal{S} = (\Pi; \mathcal{B})$ be a skeleton graph consisting of exactly one cycle C with an undefined weight function. Let α denote the weight of U_1U_2 under a realization consistent with \mathcal{S} . Then there exist a minimum possible value α_m and a maximum possible value α_M , and each value $\alpha = \alpha_m, \alpha_m + 1, \dots, \alpha_M$ appears as feasible bone weight on U_1U_2 . Finally one can provide one consistent realization for each possible weight function in polynomial time.*

Proof. When our cycle C has odd length, then Lemma 4.1 and Corollary 4.2 apply and we have nothing to prove. So assume now that $C = (U_1U_2, \dots, U_{2k}U_1)$. Applying the method of Observation 2.1 with weight 0 for all chords in the bone U_1U_2 and 1 for all other chords, the derived maximum weight perfect 1-factor provides the value α_m and a corresponding degree sequence realization. If we consider the opposite weight function then the maximum weight perfect 1-factor provides the value α_M and a corresponding degree sequence realization.

Finally one can find in polynomial time at least one realization for each value $\alpha = \alpha_m, \alpha_m + 1, \dots, \alpha_M$ applying Theorem 4.3 as follows: from this statement we know that G can be transformed into G' using F -swaps. In this setup the chords are the vertex pairs within the bones, all other vertex pairs are forbidden. When G and G' are from $\mathbb{G}(\Pi, \mathcal{B})$ then they can be consistent with different weight functions w and w' . In any procedure transforming G into G' each F -swap alters the edges and non-edges along an alternating chord circuit C' . There are two possibilities: this C' can go around the bone-circuit C zero or an even number of times or an odd number of times. In the first case for each bone the number of edges in this bone before and after the swap will be the same. In the second case the numbers of the edges within the bones increase and decrease with exactly one, alternately. Therefore each possible value α between α_m and α_M must occur in the bone U_1U_2 . \square

Now we discuss in short another possible multipartite skeleton graph:

Lemma 4.5. *Let the skeleton graph $\mathcal{S} = (\Pi; \mathcal{B}, w)$ be a tree and let L be a leaf in this tree. Assume that realization $G \in \mathbb{G}(\mathbf{d})$ is consistent with \mathcal{S} . Then the weight function w and the value $D(L)$ are completely determined by the values $D(U)$, $U \neq L$.*

Proof. This statement is almost trivial. One can argue in the same way as it happened in the proof of Lemma 4.1: starting from the leaves different from L and working along the paths toward L one can determine all $D(UW)$ values along the tree. At the last step on the unique bone UL the value $w(UL)$ must be the same as $D(L)$. \square

Putting together these statements we have the following result:

Theorem 4.6. *Let \mathbf{d} be a sequence of $|V|$ positive integers and let $\mathcal{S} = (\Pi; \mathcal{B})$ be a connected skeleton graph with at most one cycle. Then we can find in polynomial time all weight functions w for which there exists realizations of \mathbf{d} which are consistent with the skeleton graph $\mathcal{S} = (\Pi; \mathcal{B}, w)$ along with at least one realization for each possible weight function.*

Proof. We may and will assume that the skeleton graph has exactly one cycle C and trees connected to the vertices of that cycle because otherwise Lemma 4.5 would apply. If at vertex class U of C in the connected tree the vertex U has a neighbors, then we consider a disjoint subtrees, all rooted in U . For each subtree the application of Lemma 4.5 determine the corresponding weight function values, and together they determine the “residual” $D(U)$ for the cycle C . This can be done in polynomial time.

If C is odd, then it provides one unique weight function as possible setup for consistent realizations. The usual application of the Tutte method provides in polynomial time a weakly consistent realization of \mathbf{d} which will be automatically consistent.

If C is even, then the application of Lemma 4.4 provides the possible weight functions, together with actual consistent realizations for each possible weight function. \square

This finished the discussion of the existence problem of consistent realizations for skeleton graphs $\mathcal{S} = (\Pi; \mathcal{B}, w)$ where all connected components contain at most one cycle. In the remainder of this section we discuss briefly the connectivity problem of the space of all consistent realizations.

First of all we have to recognize that instead of asking the connectivity of the realization space $\mathbb{G}(\mathcal{S})$ under the regular swap operations we have to consider the F -swap operations, defined by the forbidden edges outside the active bones.

Assume that our skeleton graph is connected and it has at most one cycle. If this cycle is odd, then there is at most one weight function for which $\mathbb{G}(\mathcal{S})$ is not empty, and every weakly consistent realization will be automatically consistent as well, so the F -swap operations do not destroy the consistency. The application of Theorem 4.3 proves the connectivity of the space.

When the cycle under consideration is even then we have a more complex situation. First of all there may be several different weight functions with consistent realizations, and—similarly to the bipartite case $\mathcal{S}(k)$ —it is possible that $\mathbb{G}(\mathcal{S})$ is not connected under F -swaps. However, again similarly to the bipartite case, one can organize the F -swap sequence such that whenever we have to leave the current weight function w into w' —which differs from w with one on each bone along the cycle—then the next F -swap goes back again to the original weight function. Thus, $\mathbb{G}(\mathcal{S})$ is connected under F -swaps and “double F -swaps”.

References

- [1] Y. Amanatidis, B. Green, M. Mihail. Graphic realizations of joint-degree matrices. *Manuscript*. 2008.
- [2] G. Amanatidis, B. Green, M. Mihail. Graphic Realizations of Joint-Degree Matrices. [arXiv:1509.07076](https://arxiv.org/abs/1509.07076) 2015.
- [3] É. Czabarka, A. Dutle, P.L. Erdős, I. Miklós. On Realizations of a Joint Degree Matrix. *Discrete Appl. Math.* 181:283–288, 2015. [doi:10.1016/j.dam.2014.10.012](https://doi.org/10.1016/j.dam.2014.10.012)
- [4] É. Czabarka. On realizations of a Partition Adjacency Matrix, *unpublished manuscript* 2014.
- [5] J. Edmonds. Paths, trees, and flowers. *Can. J. Math.* 17:449–467, 1965. [doi:10.4153/CJM-1965-045-4](https://doi.org/10.4153/CJM-1965-045-4)
- [6] J. Edmonds. Maximum matchings and a polyhedron with 0,1-vertices. *Journal of Research National Bureau of Standards Section 69B*:125–130, 1965.
- [7] P. Erdős, T. Gallai. Gráfok előírt fokú pontokkal (Graphs with prescribed degree of vertices). *Mat. Lapok*, 11:264–274, 1960. (in Hungarian)
- [8] P.L. Erdős, Z. Király, I. Miklós. On the swap-distances of different realizations of a graphical degree sequence. *Comb. Prob. Comp.* 22(3):366–383, 2013. [doi:10.1017/S0963548313000096](https://doi.org/10.1017/S0963548313000096)
- [9] P.L. Erdős, S.Z. Kiss, I. Miklós, L. Soukup. Approximate Counting of Graphical Realizations. *PLOS One* #e0131300, 2015. [doi:10.1371/journal.pone.0131300](https://doi.org/10.1371/journal.pone.0131300)
- [10] M. Gjoka, B. Tillman, A. Markopoulou. Construction of Simple Graphs with a Target Joint Degree Matrix and Beyond. *Proceedings of IEEE INFOCOM* April 26 – May 1, Honkong, PR Chine 1553–1561, 2015. ISBN 978-1-4799-8381-0
- [11] S.L. Hakimi. On the realizability of a set of integers as degrees of the vertices of a graph. *SIAM J. Appl. Math.* 10:496–506, 1962. [doi:10.1137/0110037](https://doi.org/10.1137/0110037)
- [12] V. Havel. A remark on the existence of finite graphs (in Czech). *Časopis Pěst. Mat.* 80:477–480, 1955.
- [13] M.R. Jerrum, A. Sinclair, E. Vigoda. A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix with Nonnegative Entries. *Journal of the ACM* 51(4):671–697, 2004. [doi:10.1145/1008731.1008738](https://doi.org/10.1145/1008731.1008738)
- [14] Hyunju Kim, Z. Toroczkai, P.L. Erdős, I. Miklós, L.A. Székely. Degree-based graph construction. *J. Phys. A: Math. Theor.* 42:392001, 2009. [doi:10.1088/1751-8113/42/39/392001](https://doi.org/10.1088/1751-8113/42/39/392001)
- [15] Z. Király. Recognizing graphic degree sequences and generating all realizations. *EGRES Technical Report TR-2011-11* ISSN 1587-4451, 2012.
- [16] P. Mahadevan, D. Krioukov, K. Fall, A. Vahdat. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review* 36(4):135–146, 2006 [doi:10.1145/1151659.1159930](https://doi.org/10.1145/1151659.1159930)

- [17] A.N. Patrinos, S.L. Hakimi. Relations between graphs and integer-pair sequences. *Discrete Mathematics* 15:437–358, 1976. doi:[10.1016/0012-365X\(76\)90048-0](https://doi.org/10.1016/0012-365X(76)90048-0)
- [18] J. Petersen. Die Theorie der regulären Graphs. *Acta Math.* 15:193–220, 1891.
- [19] H.J. Ryser. *Combinatorial Mathematics*, Carus Math. Monograph **14**, Math. Assoc. of America, 1963.
- [20] J.K. Senior. Partitions and their Representative Graphs. *Amer. J. Math.*, 73:663–689, 1951. doi:[10.2307/2372318](https://doi.org/10.2307/2372318)
- [21] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency* Algorithms and Combinatorics 24. Springer. 2004. Chapter 26. ISBN: 978-3-540-44389-6
- [22] I. Stanton, A. Pinar. Constructing and sampling graphs with a prescribed joint degree distribution. *ACM Journal on Experimental Algorithms* 17(1): Article No. 3.5, 2012. doi:[10.1145/2133803.2330086](https://doi.org/10.1145/2133803.2330086)
- [23] L.A. Székely. *personal communication*, 2014.
- [24] W.T. Tutte. The factorization of linear graphs. *J. London Math. Soc.* 22:107–111, 1947. doi:[10.1112/jlms/s1-22.2.107](https://doi.org/10.1112/jlms/s1-22.2.107)
- [25] W.T. Tutte. The factors of graphs. *Canad. J. Math.* 4:314–328, 1952. doi:[10.4153/CJM-1952-028-2](https://doi.org/10.4153/CJM-1952-028-2)
- [26] W.T. Tutte. A short proof of the factors theorem for finite graphs. *Canad. J. Math.* 6:347–352, 1954. doi:[10.4153/CJM-1954-033-3](https://doi.org/10.4153/CJM-1954-033-3)