Combinatorics, Probability and Computing (2013) 22, 366–383.
doi:10.1017/S0963548313000096

On the Swap-Distances of Different Realizations of a Graphical Degree Sequence

PÉTER L. ERDŐS,[1†‡] ZOLTÁN KIRÁLY[2§] and ISTVÁN MIKLÓS[1‡¶]

[1]Alfréd Rényi Institute, Reáltanoda u 13-15 Budapest, 1053 Hungary
(e-mail: {erdos.peter,miklos.istvan}@renyi.mta.hu)
[2]Department of Computer Science and EGRES (MTA-ELTE), Eötvös University,
Pázmány Péter sétány 1/C, Budapest, 1117 Hungary
(e-mail: kiraly@cs.elte.hu)

Received 25 July 2012; revised 3 March 2013

One of the first graph-theoretical problems to be given serious attention (in the 1950s) was the decision whether a given integer sequence is equal to the degree sequence of a simple graph (or *graphical*, for short). One method to solve this problem is the greedy algorithm of Havel and Hakimi, which is based on the *swap* operation. Another, closely related question is to find a sequence of swap operations to transform one graphical realization into another of the same degree sequence. This latter problem has received particular attention in the context of rapidly mixing Markov chain approaches to uniform sampling of all possible realizations of a given degree sequence. (This becomes a matter of interest in the context of the study of large social networks, for example.) Previously there were only crude upper bounds on the shortest possible length of such swap sequences between two realizations. In this paper we develop formulae (Gallai-type identities) for the *swap-distance*s of any two realizations of simple undirected or directed degree sequences. These identities considerably improve the known upper bounds on the swap-distances.

AMS 2010 *Mathematics subject classification*: Primary 05C07
Secondary 05C20, 05C45

1. Introduction

The comprehensive study of graphs (or more precisely *linear graphs*, as they were called at that time) began some time in the late 1940s, in seminal works by P. Erdős, P. Turán,

† Supported in part by the Hungarian NSF, under contract NK 78439.
‡ PLE and IM acknowledge the financial support of grant #FA9550-12-1-0405 from the US Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA).
§ Supported by grants (CNK 77780 and CK 80124) from the National Development Agency of Hungary, based on a source from the Research and Technology Innovation Fund, and by TÁMOP grant 4.2.1./B-09/1/KMR-2010-0003.
¶ Supported in part by a Bolyai postdoctoral stipend and by the Hungarian NSF, under contract F61730.

W. T. Tutte, T. Gallai and others. One problem that received considerable attention was the existence of certain subgraphs of a given graph G. Such a subgraph could, for example, be a perfect matching in a (not necessarily bipartite) graph, or a Hamiltonian cycle. Generally these substructures are called *factors*. The first couple of important and rather general results of this kind were due to Tutte (in 1952) who gave necessary and sufficient conditions for the existence of f-factors [19, 20].

In cases where G is a complete graph, the f-factor problem becomes easier: we are then simply interested in the existence of a graph with a given degree sequence, and at least two solutions of different kinds were developed around 1960. One was due to Havel [10], who constructed a famous greedy algorithm to answer this *degree sequence problem*. His algorithm was based on the notion of a *swap*. It is interesting to mention the almost completely forgotten paper of Senior [18], who studied the problem of generating graphs with multiple edges but without loops: his goal was to find possible molecules with given composition but with different structures. He also discovered the swap operation, but he called it *transfusion*. The other approach was the equally famous Erdős–Gallai theorem [3], which gave a necessary and sufficient condition in the form of a sequence of inequalities. In this latter paper Havel's method was an ingredient of the proof, and the authors also observed that their result is a consequence of Tutte's f-factor theorem.

In 1962 Hakimi studied the degree sequence problem in undirected graphs with multiple edges and loops [8]. He developed an Erdős–Gallai-type result for this much simpler case, and for the case of simple graphs he rediscovered the greedy algorithm of Havel. Since then this algorithm has been referred to as the *Havel–Hakimi algorithm*.

One can already derive a polynomial-time algorithm from the general f-factor theorem of Tutte to solve the degree sequence problem, but this was not done at the time. Havel's algorithm provided a quadratic (in n) time construction method of the required graphical *realization*.

The construction of – preferably 'typical' – graphical realizations of given degree sequences has become an important problem in the past two decades, in the context of the emergence of huge networks in the social sciences, medicine, biology and internet technology, to name only a few.

Mentioning just one example here, data are collected from anonymous surveys in epidemiological studies of sexually transmitted diseases, where the individuals specify the number of different partners they have had in a given period of time, without revealing their identity. In this case, epidemiologists should construct typical contact graphs obeying the empirical degree sequence to estimate epidemiological parameters.

To construct all possible realizations of a given degree sequence is typically a very time-consuming task, since there are usually exponentially many different realizations. Here we do not consider the computationally almost hopeless isomorphism problem. In this paper the vertices are labelled (therefore distinguishable), and two isomorphic realizations where the isomorphism changes the labels are considered to be different.

The methodology to construct all possible realizations is already not self-evident: for example, the Havel–Hakimi algorithm is not strong enough to find all of them. It is also important that no particular realization should be output more than once, and, finally,

https://doi.org/10.1017/S0963548313000096 Published online by Cambridge University Press

that the waiting times between two consecutive outputs should not be too long. These concerns were addressed in [12].

However, when our goal is to find a 'typical' realization but there are exponentially many different realizations, then it is simply not feasible to generate all of the realizations and choose one randomly. One way to overcome this difficulty is to construct a good Monte Carlo Markov chain (MCMC) method. To this end we need a particular operation to traverse the space of different realizations. This operation is called a *swap*, and it is essentially the same as the operation in Havel's algorithm. We choose four vertices, for which in the induced subgraph there is a one-factor whilst another one-factor is missing, and we exchange the existing one-factor for the missing one. This clearly preserves the degree sequence.

It is interesting to recognize that (as one can learn from Erdős and Gallai) the swap operation for this purpose was originally discovered by Petersen in 1891 [16].

The problem of Erdős–Gallai-type characterization of bipartite degree sequences was studied in the mid-1950s by Gale [6] using network flow techniques. In the same year Ryser gave a direct proof for this characterization, using matrix-theoretical language [17], and showed that any particular realization of a bipartite degree sequence can be transformed, using a sequence of swaps, into any other realization. Both results were formulated on the language of directed graphs without multiple edges but with possible loops. (For the connection between bipartite and directed degree sequence problems see Section 5.) The corresponding result for simple directed graphs (no loops, no multiple edges) is due to Fulkerson [5].

Havel–Hakimi-type results are part of the folklore of bipartite degree sequences, but it is hard to find a definitive reference for them (though [21] discusses the problem in Exercise 1.4.32 and [11] provides one proof as a by-product).

For simple directed graphs, Havel–Hakimi-type results were proved by Kleitman and Wang [13] for an extension given by Kundu [14]. Swap sequences between realizations of directed graphs were rediscovered in [4]. The situation here is more complicated than in the previous cases. It is not always enough to use only two edges for a swap, and sometimes we have to use three-edge swaps. Moreover, there are two different kinds of three-edge swaps: in the *type 1* swap the three edges form an oriented C_3 and the result of the swap is the oppositely oriented triangle; in the *type 2* swap the three involved *directed edges* determine four vertices (see [13, 4]).

In [4] a weak upper bound was proved for the swap-distance of two realizations of directed degree sequences. In the proof, all three types of swaps were applied, and counted as one. However, as LaMar proved recently (in [15]), swap sequences between any two realizations may completely omit type 2 triple-swaps. In Section 5 we will strengthen this result (see Theorem 5.5). Finally, Greenhill [7] proved that for regular directed degree sequences (when all in-degrees and out-degrees are the same), all triple-swaps can be omitted (if $n > 3$). The reason for the need for some type 2 swaps in [13, 4] was simple. In the Havel–Hakimi situation (by analogy) it was sought for one swap changing a particular directed edge to a particular non-edge. However, when a transformation sequence is looked for, then this is not a requirement. As it turns out, type 2 triple-swaps are so-called non-triangular C_6-swaps (see Remark 5.1) and can be replaced by two regular swaps.

These problems have a long and lively history but we do not survey it here. We just want to point out that knowing the maximum length of the necessary swap sequences can lead to better estimates for the mixing time of Markov chains using swap operations. Up to now there have only been weak upper bounds on those lengths: they are surely shorter than twice the number of edges in the realizations (which is equal to the sum of the values in the degrees sequence). This applies to simple directed or undirected graphical degree sequences, including bipartite ones too. (See for example [4].)

The main goal of this paper is to determine a formula for the *swap-distance* (that is, the length of the possible shortest swap sequence) between any two particular realizations G_1 and G_2. Here we will prove a Gallai-type identity,

$$\mathbf{dist}(G_1, G_2) = \frac{|E(G_1)\Delta E(G_2)|}{2} - \mathbf{maxC}(G_1, G_2), \tag{1.1}$$

where Δ denotes the symmetric difference and **maxC** is a positive integer which depends on the realizations. For directed degree sequences, triple-swaps should count as 2 in the swap-distance. (For an explanation, see the definitions after Lemma 5.3.) However, as it will turn out, we only need type 1 triple-swaps. We can forbid type 2 triple-swaps while the equation does not change.

It is very important to understand that while the right side of equation (1.1) can indeed be interpreted as 'the exact value of the swap-distance', in fact **maxC** is possibly (probably) not an efficiently computable value. We think the right goal here is to find good estimates for **maxC**. We have made the first steps in this direction: see Theorems 3.8, 4.1 and 5.6.

The structure of the paper is as follows. In Section 2 we introduce the definitions and recall some known facts and algorithms. In Section 3 we prove (1.1) for undirected degree sequences. In the very short Section 4 we describe the consequences of the previous results for bipartite degree sequences. Finally, in Section 5 we discuss the problem for directed degree sequences based on further considerations on realizations of bipartite degree sequences. By the time the present paper was in its proofreading stage, the authors have learned that one of the results, namely the identity (1.1), was discovered by Bereg and Ito in 2008 ([2]).

2. Definitions and notation

Throughout the paper G denotes an undirected simple graph with vertex set

$$V(G) = \{v_1, v_2, \ldots, v_n\}$$

and edge set $E(G)$. Consider a sequence of positive integers $\mathbf{d} = (d_1, d_2, \ldots, d_n)$. If there is a simple graph G with degree sequence $\mathbf{d}$, *i.e.*, where for each i we have $d(v_i) = d_i$, then we call the sequence $\mathbf{d}$ a *graphical sequence*, and in this case we also say that G *realizes* $\mathbf{d}$.

The analogous notions for bipartite graphs are as follows. If B is a simple bipartite graph then its vertex classes will be denoted by $U(B) = \{u_1, \ldots, u_k\}$ and $W(B) = \{w_1, \ldots, w_\ell\}$, and we keep the notation $V(B) = U(B) \cup W(B)$. The *bipartite degree sequence* of B, $\mathbf{bd}(B)$

is defined as follows:

$$\mathbf{bd}(B) = \big((d(u_1),\ldots,d(u_k)),\ (d(w_1),\ldots,d(w_\ell))\big).$$

Let G be a simple graph and assume that a, b, c and d are different vertices. If G is a bipartite graph B then we also require that for $a, b \in U(B)$, $c, d \in W(B)$. Furthermore, assume that $(a,c),(b,d) \in E(G)$ while $(b,c),(a,d) \notin E(G)$. Then

$$E(G') = E(G) \setminus \{(a,c),(b,d)\} \cup \{(b,c),(a,d)\} \tag{2.1}$$

is another realization of the same degree sequence (and if G is a bipartite graph then G' remains bipartite). The operation described above is called a *swap*. This operation is used in the Havel–Hakimi algorithm, and Petersen proved [16] (and several authors later re-proved) that any realization of a degree sequence can be transformed into any another realization of the same degree sequence using only consecutive swap operations.

Since throughout the paper all graphs will be simple, from this point we will omit the word 'simple'.

A graph G whose edges are coloured by either red or blue will be called a *red–blue graph*. For a vertex v we let $d_r(v)$ and $d_b(v)$ denote the degree of vertex v in red and blue edges, respectively. This red–blue graph is *balanced* if, for each $v \in V(G)$, we have $d_r(v) = d_b(v)$.

A *circuit* in a graph G is a closed trail (each edge can be used at most once). Since the graph is simple, a circuit is determined by the sequence of the vertices $v_0,\ldots,v_t$, where $v_0 = v_t$. Note that there can also be other indices $i < j$ such that $v_i = v_j$. A circuit is called a *cycle* if it is simple, *i.e.*, for any $i < j$, $v_i = v_j$ only if $i = 0$ and $j = t$.

A circuit (or a cycle) in a balanced red–blue graph is called *alternating* if the colour of its edges alternates (*i.e.*, the colour of the edge from v_i to v_{i+1} differs from the colour of the edge from v_{i+1} to v_{i+2}, and also edges v_0v_1 and $v_{t-1}v_t$ have different colours, and consequently alternating circuits have even length).

By Euler's famous method one can easily prove the following result.

Proposition 2.1. *If G is a balanced red–blue graph then the edge set can be decomposed into alternating circuits. If B is a bipartite balanced red–blue graph then the edge set can be decomposed into alternating cycles.*

If two graphs G_1 and G_2 are different realizations of the same degree sequence, then we associate with them the following balanced red–blue graph. The vertex set is $V(G_1) = V(G_2)$ and the edge set is the symmetric difference $E(G_1)\Delta E(G_2)$. An edge is coloured red if it is in $E(G_1) - E(G_2)$, and it is coloured blue if it is in $E(G_2) - E(G_1)$.

Definition 2.2. If G is a balanced red–blue graph then let $\mathbf{maxC}_u(G)$ denote the number of the circuits in a maximum size (= maximum cardinality) alternating circuit decomposition of G. If G_1 and G_2 are two realizations of the same degree sequence then let $\mathbf{maxC}_u(G_1,G_2) = \mathbf{maxC}_u(G)$, where G is the associated balanced red–blue graph.

Definition 2.3. Let G_1 and G_2 be two given realizations of $\mathbf{d}$. Denote by $\mathbf{dist}_u(G_1, G_2)$ the length of the shortest swap sequence from G_1 to G_2.

A pair of vertices u and v will be called a *chord* if it can hold an edge. That is, for non-bipartite graphs, uv is a chord if and only if $u \neq v$, but for a bipartite graph B, uv is a chord if and only if $u \in U(B)$ and $v \in W(B)$ or *vice versa*. If a circuit $C = v_0 \dots, v_t$ is given and $v_i v_j$ is a chord, then we will also call the pair ij of subscripts a *chord*.

For directed graphs we consider the following definitions. Let $\vec{G}$ denote a directed graph (no parallel edges, no loops) with vertex set $X(\vec{G}) = \{x_1, x_2, \dots, x_n\}$ and edge set $E(\vec{G})$. We use the bi-sequence

$$\mathbf{dd}(\vec{G}) = \big((d_1^+, d_2^+, \dots, d_n^+), (d_1^-, d_2^-, \dots, d_n^-)\big)$$

to denote the degree sequence, where d_i^+ denotes the out-degree of vertex x_i while d_i^- denotes its in-degree. A bi-sequence of non-negative integers is called a *directed degree sequence* if there exists a directed graph $\vec{G}$ such that $(\mathbf{d}^+, \mathbf{d}^-) = \mathbf{dd}(\vec{G})$. In this case we say that $\vec{G}$ *realizes* our directed degree sequence.

A directed graph $\vec{G}$ is a *balanced red–blue graph* if, for every vertex, the red in-degree is the same as the blue in-degree, and moreover the red out-degree is the same as the blue out-degree. Thus, if $\vec{G}_1$ and $\vec{G}_2$ are different realizations of the same directed degree sequence, then the associated red–blue graph (defined as for the undirected case) is a balanced red–blue graph.

The definition of alternating circuit differs from the one defined for undirected graphs as follows. A circuit $v_0, \dots, v_t$ in a balanced red–blue graph $\vec{G}$ is alternating if both the colours and the directions alternate (*e.g.*, if $v_i v_{i+1}$ is a red directed edge then $v_{i+2} v_{i+1}$ is a blue directed edge).

Again, by Euler's method one can easily prove the following result.

Proposition 2.4. *If $\vec{G}$ is a balanced red–blue graph then the edge set can be decomposed into alternating circuits.*

Definition 2.5. Assume that $\vec{G}$ is a directed balanced red–blue graph and let $\mathbf{maxC}_d(\vec{G})$ denote the number of the circuits in a maximum-size alternating circuit decomposition of $\vec{G}$. If $\vec{G}_1$ and $\vec{G}_2$ are two realizations of the same directed degree sequence, then let $\mathbf{maxC}_d(\vec{G}_1, \vec{G}_2) = \mathbf{maxC}_d(\vec{G})$, where $\vec{G}$ is the associated balanced red–blue graph.

For directed graphs we use the old trick applied by Gale [6]: each directed graph $\vec{G}$ can be represented by a bipartite graph $B(\vec{G})$, where each class consists of one copy of every vertex. The edges adjacent to a vertex u_x in class U represent the out-edges from x, while the edges adjacent to a vertex w_x in class W represent the in-edges to x (so a directed edge xy is identified with the edge $u_x w_y$). Note that the directed degree sequence of $\vec{G}$ is the *same* as the bipartite degree sequence of $B(\vec{G})$. If $\vec{G}$ is a directed balanced red–blue graph then naturally we get $B(\vec{G})$ as a balanced red–blue graph, and the alternating circuits of $\vec{G}$ correspond to the alternating cycles of $B(\vec{G})$. For an alternating circuit $\vec{C}$ of $\vec{G}$ we denote the corresponding alternating cycle of $B(\vec{G})$ by C.

https://doi.org/10.1017/S0963548313000096 Published online by Cambridge University Press

As loops are not allowed in $\vec{G}$, edges of the form $u_x w_x$ are also forbidden in $B(\vec{G})$, so they will be called *non-chords*.

For two graphs G_1 and G_2 (or bipartite graphs or directed graphs) with the same degree sequence (or bipartite degree sequence or directed degree sequence, respectively) we will use $H'(G_1, G_2)$ for the *halved Hamming distance* $|E(G_1)\Delta E(G_2)|/2$. Note that $H'(G_1, G_2)$ is the same as the number of red (or blue) edges in the associated balanced red–blue graph G.

3. Undirected degree sequences

In this section we prove equality (1.1) for undirected degree sequences.

Lemma 3.1. *Let $C = v_0, v_1, \dots v_{2t} = v_0$ be an alternating circuit in a balanced red–blue graph G, in which for some $i < j < 2t$, $j - i$ is even and $v_i = v_j$. Then the circuit can be decomposed into two shorter alternating circuits.*

Proof. Since both $v_i, v_{i+1}, \dots v_j$ and $v_j, v_{j+1}, \dots v_{2t}, v_1, \dots v_i$ contain an even number of edges, both of them form alternating circuits. □

Definition 3.2. We call an alternating circuit $C = v_0, v_1, \dots v_{2t}$ *elementary* if (i) no vertex appears more than twice in it, and (ii) there exists an integer $0 \leqslant i < 2t$ such that both vertices v_i and v_{i+1} occur only once in the circuit.

Lemma 3.3. *Let $C_1, \dots, C_h$ be a maximum-size alternating circuit decomposition of a balanced red–blue graph G (that is, $h = \mathbf{maxC}_u(G)$). Then each circuit is elementary.*

Proof. **(i)** First assume that a circuit $C_z = v_0, \dots, v_{2t}$ contains the vertex v three times. Then two of the occurrences have the same parity, and Lemma 3.1 applies. But this is contradiction to the maximality. Therefore any vertex in any circuit of a maximum-size decomposition occurs at most twice, and the two subscripts of each repeated vertex (within any circuit) have different parities. We call a pair $0 \leqslant i < j < 2t$ a *non-chord* if $v_i = v_j$. The *length* of a non-chord ij is defined to be $\min(|i - j|,\ 2t - |i - j|)$. We have proved that each index i is a part of at most one non-chord, and the length of any non-chord is odd.

We next prove that non-chords cannot intersect. More precisely, if $0 \leqslant i < k < j < \ell < 2t$, then if ij is a non-chord then $k\ell$ is a chord. Let C' be the following alternating circuit:

$$v_0, \dots, v_i = v_j, v_{j-1}, v_{j-2}, \dots, v_k, \dots v_{i+1}, v_i = v_j, v_{j+1}, \dots, v_\ell \dots v_{2t}.$$

In this new circuit (which consists of the same edges as the original circuit) the new index of vertex v_k is k', where $k - k'$ is odd. Therefore, if $k - \ell$ was odd then $k' - \ell$ is even. Thus, for this circuit Lemma 3.1 applies, which in turn shows that the original circuit decomposition is not a maximum-size one, a contradiction.

(ii) By re-indexing the vertices of the circuit we may assume that $0k$ is the shortest non-chord of C_z. Since G is simple we have $k > 2$. Consequently indices 1 and 2 cannot participate in any non-chord, otherwise they would induce crossing non-chords. □

From the middle part of the proof one can deduce a much stronger statement. Given a circuit $C = v_0, \dots, v_{2t}$, we call a vertex *unique* if it appears exactly once in that circuit.

Theorem 3.4. *Let $C_1, \dots, C_h$ be a maximum-size alternating circuit decomposition of a balanced red–blue graph G. Then*

(i) *each circuit $C_z = v_0, \dots, v_{2t}$ contains at least $2t/3 + 2$ unique vertices,*
(ii) *the length of each circuit in the decomposition is at most $(3/2)(n-1)$, and consequently*

$$\mathbf{maxC}_u(G) \geqslant \left\lceil \frac{2|E(G)|}{3n} \right\rceil.$$

Proof. **(i)** Let μ denote the number of unique vertices of the circuit and let ν denote the number of non-unique vertices (not indices). Since no vertex appears more than twice in the circuit, $2t = \mu + 2\nu$ and the number of non-chords is exactly ν.

If $t = 2$ then non-chords do not exist, so there is nothing to prove. Assume now that $t > 2$, and consider the following planar graph P with $2t$ vertices. First we draw a convex $2t$-gon on the plane with vertices $p_0, \dots, p_{2t-1}$. Next, for each non-chord ij we connect p_i to p_j by a straight-line segment. The proof of Lemma 3.3 shows that there are no crossing non-chords, so this is a planar embedding.

Now we take the planar dual P^* and delete the vertex of the dual corresponding to the infinite face of P. We call the resulting graph T.

It is easy to see that T is a tree. Indeed, we can argue by contradiction. If T contains a cycle then the original planar graph contains a vertex v within this cycle. But each original vertex of the graph is next to the ocean. Therefore vertex v is also next to the ocean, so the dual vertex O corresponding to the ocean must not be outside C. But that would imply that O belongs to the cycle, a contradiction.

The edges of T correspond to the non-chords, so $|E(T)| = |V(T)| - 1 = \nu$. The vertices of the tree correspond to the finite faces of P. We claim that if $v \in V(T)$ has degree at most 2 in the tree, then the corresponding face contains at least 2 unique vertices. If a face is adjacent to one non-chord (it corresponds to a leaf in T) then it has at least two unique vertices since G is simple. Suppose there is a face of P adjacent to two non-chords, ij and $i'j'$, where we may assume that $i < i' < j' < j$. As G is simple and the non-chords have odd length, we can conclude that $i' - i + j - j' \geqslant 4$, proving the claim.

Let $n_{\leqslant r}$ (and $n_{\geqslant r}$) denote the number of vertices of T having degree at most r (at least r, respectively). It can be proved by induction on $|V(T)|$ that if $|V(T)| > 1$ then $n_{\leqslant 1} \geqslant n_{\geqslant 3} + 2$. (If $|V(T)| \geqslant 3|$ and we delete a leaf ℓ, then either none of $n_{\leqslant 1}$ and $n_{\geqslant 3}$ is changed, if the neighbour of ℓ originally has degree two, or $n_{\leqslant 1}$ decreases by exactly one and $n_{\geqslant 3}$ decreases by at most one.) Consequently $n_{\leqslant 2} \geqslant n_{\geqslant 3} + 2$. So

$$\mu \geqslant 2n_{\leqslant 2} \geqslant |V(T)| + 2 = |E(T)| + 3 = \nu + 3.$$

https://doi.org/10.1017/S0963548313000096 Published online by Cambridge University Press

Since $2t = \mu + 2v$, we have $2t \leqslant 3\mu - 6$, and consequently $\mu \geqslant 2t/3 + 2$, proving our first statement.

(ii) To prove the second statement we only need a simple calculation,

$$t + (t/3 + 1) \leqslant \mu/2 + v + \mu/2 = \mu + v \leqslant n,$$

so really $2t \leqslant (3/2)(n-1)$. □

Now we are ready to analyse the minimum-size swap sequences. We start with the simplest case.

Lemma 3.5. *Assume that G_1 and G_2 are two realizations of the same degree sequence, and G is a balanced red–blue graph consisting of the edges in $E(G_1)\Delta E(G_2)$. Suppose that $E(G)$ is one alternating elementary circuit C of length $2t$. Then there is a swap sequence of length $t-1$ between G_1 and G_2.*

Proof. Let us call G_1 the *start* graph and G_2 the *stop* graph. We apply induction on the size of the symmetric difference $|C|$ of the actual start and stop graphs. We may assume that v_0 occurs exactly once in C and the current v_0v_1 edge belongs to the start graph (since this circuit is elementary, due to Lemma 3.3, we can always renumber the vertices accordingly).

When $t = 2$ then the statement is clear, since C is an alternating cycle of length 4, so assume now that $t > 2$. Consider the chords v_0v_1, v_1v_2, v_2v_3, v_3v_0. (Recall that by definition we have $v_0v_1, v_2v_3 \in E(G_1) \setminus E(G_2)$ and $v_1v_2 \in E(G_2) \setminus E(G_1)$, while $v_3v_0 \notin E(G_1)\Delta E(G_2)$.)

When chord v_3v_0 is a non-edge in the start graph (and therefore in the stop graph), then we can perform the $v_0v_1, v_2v_3 \Rightarrow v_1v_2, v_3v_0$ swap in the start graph. After this operation the circuit will be shorter by two edges and remain elementary. So we can apply the inductive hypothesis for the new start/stop graph pair. If, however, the chord v_3v_0 is an edge in both the start and stop graphs, then we can carry out the $v_1v_2, v_3v_0 \Rightarrow v_0v_1, v_2v_3$ swap in the stop graph, still maintaining all the necessary properties. So we can proceed with the induction on the new start/stop graph pair. □

Theorem 3.6. *For all pairs of realizations G_1, G_2 of the same degree sequence, we have*

$$\mathbf{dist}_u(G_1, G_2) = H'(G_1, G_2) - \mathbf{maxC}_u(G_1, G_2). \tag{3.1}$$

Proof. **(i(a))** The inequality left-hand side $\leqslant$ right-hand side is a simple application of Lemma 3.5: take a maximal alternating circuit decomposition $C_1, \ldots, C_k$, where $k = \mathbf{maxC}_u(G_1, G_2)$, and define realizations $G_1 = H_0, H_1, \ldots, H_{k-1}, H_k = G_2$ such that, for all $i = 0, \ldots, k-1$, realizations H_i and H_{i+1} differ exactly in C_i. Then by Lemma 3.3 all circuits are elementary, so the application of Lemma 3.5 for each pair H_i, H_{i+1} proves this inequality.

(i(b)) One can find a recursive proof too. This is based on the following easy observation. Assume that the shortest circuit C_1 in the previous maximal decomposition has the

https://doi.org/10.1017/S0963548313000096 Published online by Cambridge University Press

shortest length among all circuits in all possible maximal circuit decompositions. Then we have the following lemma.

Lemma 3.7. *There exists no edge in any of the other circuits that would divide C_1 into two odd length trails.*

Proof Proof of Lemma 3.7. Assume the opposite, *i.e.*, the chord $v_1, v_{2\ell}$ of C_1 is an edge in C_2. Then this edge together with one of the trails of C_1 forms a shorter circuit than C_1, while the other trail together with the remaining part of C_2 forms another alternating circuit. So we have constructed another circuit decomposition with the same number of circuits but a shorter shortest circuit, a contradiction. (It is still possible for a chord in C_1 to belong to another circuit too, but this divides C_1 into two even-length trails. However, this will not cause any problem.) □

Now we can operate as follows. Consider the (actual) symmetric difference, and find a maximal circuit decomposition with a shortest elementary circuit. Apply the procedure in Lemma 3.5 for this circuit (permissible by Lemma 3.7). Repeat the whole process with the new (and smaller) symmetric difference.

(ii) We finish the proof of Theorem 3.6 by proving that left-hand side ⩾ right-hand side. We rearrange (3.1) into the form

$$\mathbf{maxC}_u(G_1, G_2) \geqslant H'(G_1, G_2) - \mathbf{dist}_u(G_1, G_2).$$

Assume that the sequence $G_1 = H_0, H_1, \ldots, H_{k-1}, H_k = G_2$ describes a minimum-length realization sequence from G_1 to G_2, where for each $i = 0, \ldots, k-1$ the graphs H_i and H_{i+1} are in swap-distance 1. It is clear that any consecutive swap subsequence from H_i to H_j must also be a minimum one. For each i we use the notation

$$\Delta_i := E_1 \Delta E(H_i).$$

We are going to construct a circuit decomposition of $E_1 \Delta E_2 = \Delta_k$ into $\geqslant H'(G_1, G_2) - \mathbf{dist}_u(G_1, G_2)$ alternating circuits. By part (i) it will also prove that the two sides are actually equal (otherwise the swap sequence cannot be minimum). We proceed with induction: we will show that for all $i = 0, \ldots, k$ we have

$$\mathbf{maxC}_u(G_1, H_i) \geqslant H'(G_1, H_i) - \mathbf{dist}_u(G_1, H_i). \tag{3.2}$$

For $i = 0$ this is clearly true; if $i = k$ then the main statement is proved. Now we assume (3.2) for subscript i and we are going to prove it for $i + 1$. By the hypothesis we know that $\mathbf{dist}_u(G_1, H_i) = i$. We are going to distinguish cases depending on the relations between $E(H_i) \Delta E(H_{i+1}) = S$ and Δ_i.

Assume first that $|S \cap \Delta_i| = 0$. Then the number of circuits in the decomposition of Δ_{i+1} is increased by one (compared to the maximum decomposition of Δ_i), the number of edges is increased by four, and finally the number of swaps is increased by one again. Inequality (3.2) is maintained.

Now assume that $|S \cap \Delta_i| = \ell > 0$. Since S is derived from the swap transforming H_i into H_{i+1}, the two existing edges among the four chords defining S are edges in H_i

rather than edges in H_{i+1}, and the analogous statement is true for the two missing edges. Therefore the chords in $S \cap \Delta_i$ are in the same states in H_0 and in H_{i+1}, and then we obtain one of the following options.

(a) If $\ell = 1$ then this chord does not belong to Δ_{i+1}, so the other three chords of S extend the original circuit. Therefore the number of circuits is the same as before, while $|\Delta_{i+1}| = |\Delta_i| + 2$ and the number of necessary swaps is increased by one. Inequality (3.2) is maintained.

(b) If $\ell > 1$ then the ℓ common chords can be in at most ℓ circuits. It can happen that some circuits meld into a smaller number of circuits after the swap on S is performed, but this can decrease the number of circuits with at most $\ell - 1$. Furthermore, $|\Delta_{i+1}| = |\Delta_i| + 4 - 2\ell$, so the number of necessary swaps is ultimately increased by one. Inequality (3.2) is maintained.

The proof of Theorem 3.6 is complete. □

As mentioned earlier, the value $\mathbf{maxC}_u$ does not seem efficiently computable. Therefore Theorem 3.6 does not directly help us to find a shortest swap sequence between two particular realizations. However, good upper and lower bounds on this value may be useful. This is clear, however, that these bounds depend not only on the number of the edges (which is $|E|$) in one realization but on the size of the symmetric difference. When $|E|$ is small, say $|E| \leqslant \frac{1}{2}\binom{n}{2}$, then the size of the symmetric difference can be as big as $2|E|$. If $|E|$ is much higher then the symmetric difference becomes small.

Assume now that the graphical degree sequence under investigation is $(1, 1, 1, \ldots, 1)$. All realizations are perfect matchings, and if two of them form one alternating Eulerian cycle, then the actual swap-distance, by Theorem 3.6, is $|E| - 1$. This is just half of the old estimate. The consequence is that the diameter of the corresponding Markov chain can be as big as $|E| - 1$.

Next we give a general bound on the swap-distance (which is in some sense sharp), and then we formulate some conjectures. For a given degree sequence $\mathbf{d} = \{d_1, d_2, \ldots, d_n\}$, let m denote $(\sum d_i)/2$, the number of edges in any realization, and let m^* denote $\sum \min(d_i, n - d_i)$, an upper bound on the number of edges in a balanced red–blue graph associated with two realizations G_1 and G_2.

Theorem 3.8. *For all pairs of realizations G_1, G_2 of the same degree sequence of length n, we have*

$$\mathbf{dist}_u(G_1, G_2) \leqslant H'(G_1, G_2) \cdot \left(1 - \frac{4}{3n}\right)$$
$$\leqslant m^* \left(\frac{1}{2} - \frac{2}{3n}\right) \leqslant m\left(1 - \frac{4}{3n}\right).$$

Proof. This is a simple calculation using Theorem 3.4 and the simple fact that $H'(G_1, G_2) \leqslant m^*/2 \leqslant m$. □

Conjecture 3.9. *Let G be a balanced red–blue graph with n vertices and m edges. Then* (i) *there exists an alternating circuit of length at most $3n^2/m$, and* (ii) $\mathbf{maxC}_u(G) \geqslant m^2/(6n^2)$.

Such an upper bound would provide a lower bound on the distance, and could thus be useful in practical applications.

Conjecture 3.10. *For a degree sequence $\mathbf{d} = \{d_1, d_2, \ldots, d_n\}$, let m again denote $(\sum d_i)/2$, the number of edges in any realization, and let m^* denote $\sum \min(d_i, n - d_i)$. Then we conjecture the following statements:*

(i) $\mathbf{dist}_u(G_1, G_2) \leqslant H'(G_1, G_2) \cdot (1 - m/(3n^2))$,
(ii) $\mathbf{dist}_u(G_1, G_2) \leqslant m^*(1/2 - m/(6n^2))$,
(iii) $\mathbf{dist}_u(G_1, G_2) \leqslant m(1 - m/(3n^2))$,
(iv) $\mathbf{dist}_u(G_1, G_2) \leqslant 5n^2/24$.

4. Undirected bipartite degree sequences

It is easy to see that for bipartite degree sequences Theorem 3.6 applies without any changes (note that in the proof we only used chords of odd length, so they are also chords in the bipartite case). Moreover, since there is no odd cycle in a bipartite graph, the circuits in the maximum-size alternating circuit decomposition of the symmetric difference of two realizations are cycles. As a consequence, for two realizations B_1 and B_2 of a bipartite degree sequence, we can interpret $\mathbf{maxC}_u(B_1, B_2)$ as the maximum number of cycles in a decomposition into alternating cycles (which always exists) of the associated balanced red–blue graph B. However, we think that even for bipartite realizations the determination of $\mathbf{maxC}_u$ might be hard.

Let $\mathbf{bd} = \big((a_1, \ldots, a_k), (b_1, \ldots, b_\ell)\big)$ be a given bipartite degree sequence; we assume $\ell \leqslant k$. Let $n = k + \ell$, $n' = 2\ell$, $m = \sum a_i$, and let m^* denote $2\sum \min(a_i, \ell - a_i)$, an upper bound on the number of edges in a balanced red–blue graph associated with two realizations B_1 and B_2. Using that any alternating cycle has length at most n', similarly to Theorem 3.8, we get the following.

Theorem 4.1. *For all pairs of realizations B_1, B_2 of the same degree sequence* **bd** *we have*

$$\begin{aligned}\mathbf{dist}_u(B_1, B_2) &\leqslant H'(B_1, B_2) \cdot (1 - 2/n') \\ &\leqslant m^*(1/2 - 1/n') \leqslant m(1 - 2/n').\end{aligned}$$

5. Directed degree sequences

In this section we discuss directed degree sequences. We will apply the machinery of Section 4 to solve the directed degree sequence problem, using the bipartite graph $B(\vec{G})$ defined in Section 2. However, in so doing we may face a serious problem: since no loop is allowed in $\vec{G}$, we cannot use edges of form $u_x w_x$ in the process. Recall that these pairs are called non-chords. We first analyse the alternating cycles occurring during the process.

Let $\vec{G}$ be a directed balanced red–blue graph associated with two realizations $\vec{G}_1$ and $\vec{G}_2$ of the same directed degree sequence, let $B = B(\vec{G})$ be the corresponding bipartite balanced red–blue graph, and let $\vec{C}$ be an alternating circuit in $\vec{G}$ (recall that C denotes the corresponding alternating cycle in B). In this section we will mainly use this notation for the bipartite representation B, but, where relevant, we note in italics and in parentheses the corresponding notions in the original directed graph.

Case 1. Let us start with the case when there exists a vertex u_x in the cycle C such that w_x is not contained in C (of course, by symmetry, the case when w_x is in C but u_x is not can be handled in the same way). (*This is equivalent to saying that circuit $\vec{C}$ contains x only once.*) We can work with u_x at each step of the process described in Lemma 3.5. We take the trail of length 3 starting at u_x, and interchange the start and stop graphs if the first edge belongs to the stop graph (we will repeat this step routinely, observing that if we are given a swap sequence from one graph to another graph, then the reverse sequence transforms the second graph into the first one). At every step the vertex u_x remains in the cycle and w_x will not become a vertex of the cycle.

Case 2. Next assume that for each vertex $u_x \in C$ we also have $w_x \in C$, but also assume that we have a vertex u_x in C such that the trail of length 3 (along the ordering of the cycle) starting at u_x does not end at w_x. Then we can use vertex u_x in the process as before. After the first swap, there will be two vertices which will occur without their non-chord pairs in the new cycle. So we are back to Case 1.

Case 3. Finally assume that neither Case 1 nor Case 2 applies. We can handle this case as follows. Assume first that C is long enough, that is, $|C| \geqslant 8$. As every vertex participates in one non-chord, one of the two different vertices that are of distance 3 (along the cycle) from any fixed vertex u_x must differ from w_x. So we are back to Case 2 after reversing the description of C, and possibly interchanging the start and stop graphs. From now on we will call these typical swaps C_4-*swaps*.

When our cycle is of length 6, then no such trick works. (*In $\vec{G}_1$ we have an oriented triangle and $\vec{G}_2$ is identical to $\vec{G}_1$, except that it contains the other orientation of the same triangle. Then we have to use a new type of swap: we exchange the first oriented triangle with the second one.*) This means that in the bipartite graph we swap a C_6 with three non-chords in one step. For obvious reasons we will call this new swap a *triangular C_6-swap*, and the cycle itself is called a *triangular C_6-cycle*.

Remark 5.1. We can now describe the type 2 triple-swaps mentioned in Section 1 and introduced in [13, 4]: they are simply non-triangular C_6-swaps that can be implemented by two C_4-swaps.

Lemma 5.2. *If C is a cycle in the decomposition that is not a triangular C_6-cycle, then we can always perform the next swap without producing a new triangular C_6-cycle.*

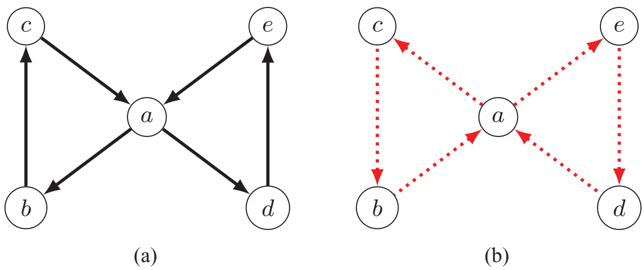

Figure 1. (Colour online) Two realizations: (a) $\vec{G}_1$, (b) $\vec{G}_2$. Vertices $X = \{a,b,c,d,e\}$ and $\mathbf{dd} = ((2,1,1,1,1), (2,1,1,1,1))$.

Proof. This is clearly the case when we are in Case 1. If we are in Case 3 or Case 2, then $|C| \geqslant 8$, and after the first swap two neighbouring vertices are deleted from the cycle, with the result that we are back to Case 1 (two non-chords disappear). □

It is important to recognize that sometimes triangular C_6-swaps are absolutely necessary: for example, let $n \geqslant 2$ be an integer and consider the following $n+1$-element directed degree sequence: $\mathbf{dd} = \big((n,n,\ldots,n,n-1,n-1,n-1), (n,n,\ldots,n,n-1,n-1,n-1)\big)$. It is clear that there are exactly two different realizations of this directed degree sequence: *i.e.*, this is a complete directed graph on $n+1$ vertices minus one oriented triangle, and this oriented triangle can be of two different kinds. For these realizations there is exactly one possible swap: the triangular C_6-swap on the six vertices of the $B(\vec{G}_i)$ realizations. (The simplest such example is $((1,1,1),(1,1,1))$.)

With these observations we have just proved that any realization of a directed degree sequence can be transferred to any other realization of the same degree sequence using only C_4-swaps and triangular C_6-swaps. Therefore, in contrast to [13] and [11], from now on we allow only these two types of swaps, while three-edge swaps of type 2 are excluded.

Next we are going to analyse the structure of the triangular C_6-cycles in a maximal cycle decomposition with a minimal number of triangular C_6-cycles. (*Let us start with the example shown in Figure 1: in this directed degree sequence the realizations consists of two oriented triangles, sharing one vertex.*)

Figure 2 shows the bipartite representation of the symmetric difference of the corresponding B_1 and B_2. It is easy to see that there are two possible cycle decompositions of this symmetric difference: one consists of two triangular C_6-cycles, but the other contains none (see Figure 3).

It is fortunate that this is the typical behaviour. We say that two cycles in the decomposition of the bipartite representation are *kissing* if there exists a vertex $x \in X$ such that both alternating cycles in the decomposition contain both u_x and w_x. If one or both kissing cycles are triangular C_6-cycles then we can transform these two cycles into a new decomposition without any triangular C_6. To that end we consider the four trails defined by u_x and w_x and pair them up in the right way.

With this observation we have just proved the following structural property.

https://doi.org/10.1017/S0963548313000096 Published online by Cambridge University Press

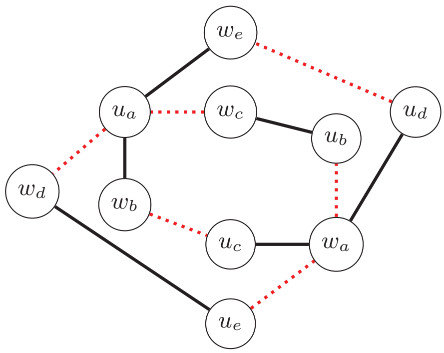

Figure 2. (Colour online) The bipartite representation of the symmetric difference.

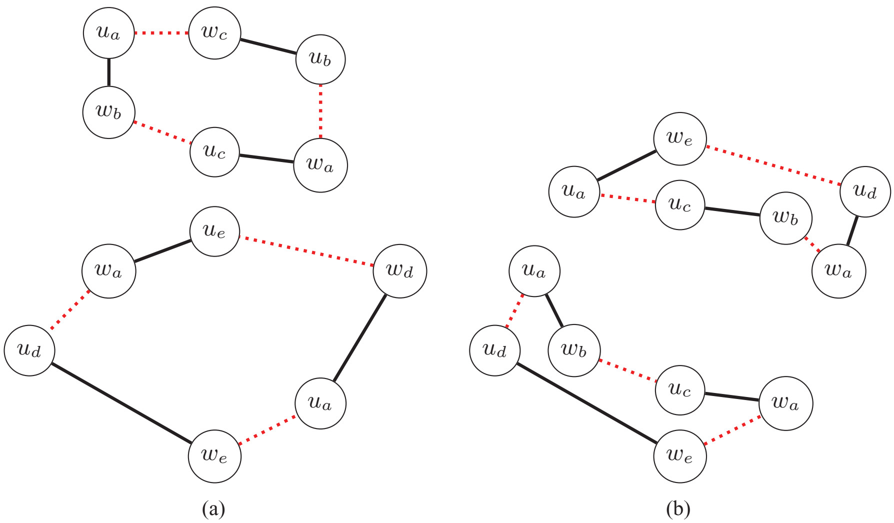

Figure 3. (Colour online) Two possible cycle decompositions: (a) into triangular C_6s, (b) into non-triangular C_6s.

Lemma 5.3. *Assume that the alternating cycle decomposition $\mathcal{C}$ of the symmetric difference of $B(\vec{G}_1)$ and $B(\vec{G}_2)$ is a maximal one with a minimum number of triangular C_6-cycles. Then no triangular C_6-cycle kisses any other cycle.*

We are now ready to define the swap-distance of two arbitrary realizations of the same directed degree sequence. We consider a *weighted swap-distance*: an ordinary C_4-swap weighs one, but a triangular C_6-swap weighs two. (This convention is well supported by the fact that two kissing triangular cycles can be transformed into two ordinary length 6 cycles, each transformable using two C_4-swaps.) So $\mathbf{dist}_d(\vec{G}_1, \vec{G}_2)$ denotes the minimum total weight of a swap sequence transforming $\vec{G}_1$ into $\vec{G}_2$. The definition of $\mathbf{maxC}_d(\vec{G}_1, \vec{G}_2)$ is analogous to the undirected case: this is the possible maximum number of directed cycles in an alternating directed cycle decomposition of the symmetric difference of the edge sets. Using these definitions, we have the following result for the minimum directed swap-distance.

Theorem 5.4. *Let* **dd** *be a directed degree sequence with realizations* $\vec{G}_1$ *and* $\vec{G}_2$. *Then*

$$\mathbf{dist}_d(\vec{G}_1, \vec{G}_2) = H'(\vec{G}_1, \vec{G}_2) - \mathbf{maxC}_d(\vec{G}_1, \vec{G}_2). \tag{5.1}$$

Proof. We can prove that the left-hand side is at most as big as the right-hand side by recalling the proof from Theorem 3.6 for bipartite graphs, taking into consideration the previous observations. By Lemma 5.2, we do not create any new triangular C_6-cycle.

Now consider the equivalent form of inequality (3.2). Fix a suitable swap sequence of minimal weighted length and assume that this contains the smallest possible number of triangular C_6-swaps among all such sequences. Let $B(\vec{G}_1) = H_0, H_1, \ldots, H_{k-1}, H_k = B(\vec{G}_2)$ denote the bipartite graph sequence which consists of the consecutive realizations generated by this swap sequence (then for each $i = 0, \ldots, k-1$ the graphs H_i and H_{i+1} are in swap-distance 1 or 2, depending on whether the swap was a simple C_4-swap or a triangular C_6-swap). It is clear that:

(A) Any consecutive swap subsequence from H_i to H_j must be also a minimum one. Furthermore, it must contain the smallest possible number of triangular C_6-swaps among all such subsequences.

For each i we use the notation

$$\Delta_i := E(H_0)\Delta E(H_i).$$

We will now revisit the proof of Theorem 3.6(ii) and try to mimic it. Whenever the swap under investigation is a triangular C_6-swap, then it cannot share a non-chord with any previously generated cycle. (This comes from a simple straightforward generalization of Lemma 5.3.)

Whenever the swap is a regular C_4-swap, we can proceed as in the proof of Theorem 3.6(ii). This concludes the proof of Theorem 5.4. □

We can strengthen the recent result by LaMar [15] as follows.

Theorem 5.5. *If we modify the definition of weighted swap-distance between two realizations of a directed degree sequence such that any length* 6 *circuit can be swapped in one step, and their weights are* 2, *then Theorem 5.4 still holds, and there exists a shortest swap sequence between any two realizations,* $\vec{G}_1$ *and* $\vec{G}_2$, *of the same directed degree sequence which contains only* C_4*-swaps and triangular* C_6*-swaps.*

The novelty here is that there always exists a shortest swap sequence which conforms to LaMar's idea.

Proof. Any length 6 circuit which is not a triangular C_6-cycle falls into Case 1 or Case 2 above, and can be transformed with two C_4-swaps whose summed weight is also 2. □

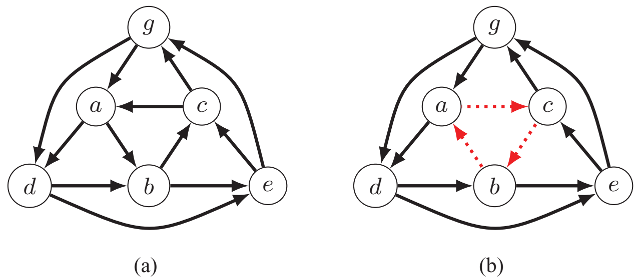

Figure 4. (Colour online) Two realizations with a triangular C_6 as symmetric difference: (a) $\vec{G}_1$, (b) $\vec{G}_2$.

Theorem 4.1 transforms into the following. Let there be given a directed degree sequence $\mathbf{dd} = \big((d_1^+,\dots,d_n^+), (d_1^-,\dots,d_n^-)\big)$. Let

$$m = \sum d_i^+ \quad \text{and} \quad m^* = \sum_i \big[\min\big(d_i^+, n-d_i^+\big) + \min\big(d_i^-, n-d_i^-\big)\big],$$

where m^* is an upper bound on the number of edges in a balanced red–blue graph associated with two realizations $\vec{G}_1$ and $\vec{G}_2$. Then we have the following result.

Theorem 5.6. *For all pairs of realizations $\vec{G}_1, \vec{G}_2$ of the same degree sequence* **dd**, *we have*

$$\begin{aligned}\mathbf{dist}_d(\vec{G}_1,\vec{G}_2) &\leqslant H'(\vec{G}_1,\vec{G}_2)\cdot\left(1-\frac{1}{n}\right)\\ &\leqslant m^*\left(\frac{1}{2}-\frac{1}{2n}\right) \leqslant m\left(1-\frac{1}{n}\right).\end{aligned}$$

To conclude our paper, recall that Greenhill proved in [7, Lemma 2.2] that for regular degree sequences any two directed realizations can be transformed into each other using only C_4-swaps. (In this case she calls these C_4-swaps *switches*.) A similar notion was studied by Berger and Müller-Hannemann (see [1]). However, consider the following example. Let **dd** be a two-regular directed degree sequence with six vertices. In Figure 4 we show two realizations. The symmetric difference of these two realizations is one triangular C_6-cycle. Therefore the swap sequence generated by Greenhill cannot be a minimal one. (Of course, in her application this was never a requirement: she uses the above-mentioned result successfully to prove rapid mixing time of the sampling algorithm for regular directed bipartite graphs.)

References

[1] Berger, A. and Müller-Hannemann, M. (2010) Uniform sampling of digraphs with a fixed degree sequence. In *Graph Theoretic Concepts in Computer Science*, Vol. 6410 of *Lecture Notes in Computer Science*, Springer, pp. 220–231.
[2] Bereg, S. and Ito, H. (2008) Transforming graphs with the same degree sequence. In *KyotoCGGT 2007, LNCS 4535* (H. Ito *et al.*, eds), pp. 25–32.
[3] Erdős, P. and Gallai, T. (1960) Graphs with prescribed degree of vertices (in Hungarian). *Mat. Lapok* **11** 264–274.

[4] Erdős, P. L., Miklós, I. and Toroczkai, Z. (2010) A simple Havel–Hakimi type algorithm to realize graphical degree sequences of directed graphs. *Electron. J. Combin.* **17** #R66.
[5] Fulkerson, D. R. (1960) Zero-one matrices with zero trace. *Pacific J. Math.* **10** 831–836.
[6] Gale, D. (1957) A theorem on flows in networks, *Pacific J. Math.* **7** 1073–1082.
[7] Greenhill, C. (2011) A polynomial bound on the mixing time of a Markov chain for sampling regular directed graphs. *Electron. J. Combin.* **18** #P234.
[8] Hakimi, S. L. (1962) On the realizability of a set of integers as degrees of the vertices of a simple graph. *SIAM J. Appl. Math.* **10** 496–506.
[9] Hakimi, S. L. (1965) On the degrees of the vertices of a directed graph. *J. Franklin Institute* **279** 290–308.
[10] Havel, V. (1955) A remark on the existence of finite graphs (in Czech). *Časopis Pěst. Mat.* **80** 477–480.
[11] Kim, H., Toroczkai, Z., Erdős, P. L., Miklós, I. and Székely, L. A. (2009) Degree-based graph construction. *J. Phys. A: Math. Theor.* **42** 392001.
[12] Király, Z. (2012) Recognizing graphic degree sequences and generating all realizations. EGRES Technical Report TR-2011-11. `http://www.cs.elte.hu/egres`
[13] Kleitman, D. J. and Wang, D. L. (1973) Algorithms for constructing graphs and digraphs with given valences and factors. *Discrete Math.* **6** 79–88.
[14] Kundu, S. (1973) The k-factor conjecture is true. *Discrete Math.* **6** 367–376.
[15] LaMar, M. D. (2011) Directed 3-cycle anchored digraphs and their application in the uniform sampling of realizations from a fixed degree sequence. In *ACM & IEEE & SCS Proc. 2011 Winter Simulation Conference* (S. Jain, R. R. Creasey *et al.*, eds), pp. 1–12.
[16] Petersen, J. (1891) Die Theorie der regulären Graphen. *Acta Math.* **15** 193–220.
[17] Ryser, H. J. (1957) Combinatorial properties of matrices of zeros and ones. *Canad. J. Math.* **9** 371–377.
[18] Senior, J. K. (1951) Partitions and their representative graphs. *Amer. J. Math.* **73** 663–689.
[19] Tutte, W. T. (1952) The factors of graphs. *Canad. J. Math.* **4** 314–328.
[20] Tutte, W. T. (1954) A short proof of the factors theorem for finite graphs. *Canad. J. Math.* **6** 347–352.
[21] West, D. B. (2001) *Introduction to Graph Theory*, second edition, Prentice Hall.