Degree-based construction and sampling of simple graphs

Zoltán Toroczkai
Department of Physics and iCeNSA, U. Notre Dame

with:


Hyunju Kim (Virginia Tech)


Charo I. Del Genio (MPIPKS)


Kevin E. Bassler (U. Houston)


Péter L. Erdős (Rényi Inst)


László A. Székely (U South Carolina)


István Miklós (Rényi Inst)

Fundamentals

In network modeling one often needs to generate graphs without having full connectivity information.

There are cases where only the degree sequence ($\mathcal{O}(N)$) is available in form of data.

Examples include

1) Data generated from surveys for epidemic studies.

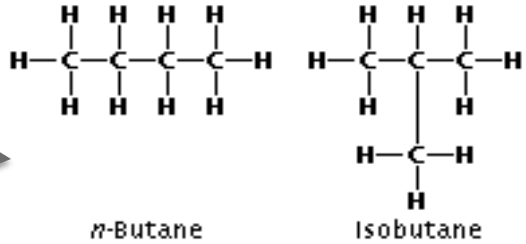
R. Anderson & R. May, *Nature* **333**, 514 (1988).

F. Liljeros, C. Edling, L.A.N. Amaral, H. E. Stanley, Y. Åberg, *Nature* **411**, 907908, (2001) .

2) Chemistry. Note: node = atom, edge = bond, degree = valence

2a) Structural isomers of alkenes.

- e.g., C_4H_{10}, butane has 2 isomers
- $CH_3(CH_2)_6CH_3$ has 18 isomers
- $C_{20}H_{42}$ has 366,319 isomers.



2b) Topological indices of molecular branching. *The **Randič index***.

Problem: Given a graphical sequence of integers $\boldsymbol{d}$, find a simple graph $G(V,E)$ with degree sequence $\boldsymbol{d}$ which maximizes:

$$R_\alpha(G) = \sum_{(i,j)\in E} (d_i d_j)^\alpha$$

The case $\alpha = -1/2$ is the classical Randič index.

Strongly correlates with physical properties such as "boiling points of hydrocarbons and the retention volumes and retention times obtained from chromatographic studies".

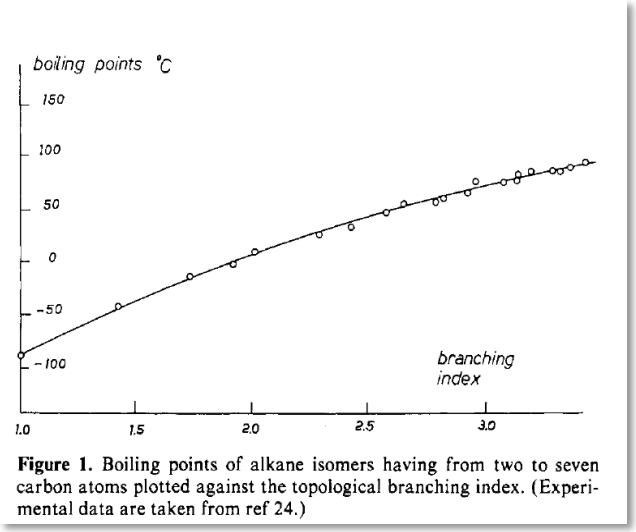


Figure 1. Boiling points of alkane isomers having from two to seven carbon atoms plotted against the topological branching index. (Experimental data are taken from ref 24.)

M. Randič, "On characterization of molecular branching", *J. Amer. Chem. Soc.* 97, 6609 (1975).

General Motivation: understand how and to what level processes on networks (information flow, epidemics, etc.) are affected by ***the degree sequence alone*** and nothing else.

Hence we need to be able to build ensembles of graphs realizing a given degree sequence and sample from them with *known distributions*.

[note: sequence, not distribution]

Degree sequences

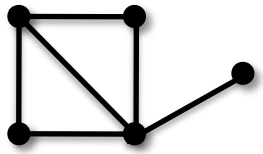
(we'll talk about labeled graphs, only)

Given a graph G, the set of non-negative integers $\boldsymbol{d}(G) = \boldsymbol{d} = \{d_1, d_2, \ldots, d_n\}$ forms its *degree sequence*.

Not all sequences of non-negative integers will form the degree sequence of a ***simple graph!*** (Graph without loops and multiple edges between the same pair of nodes).

Examples:

1) $\boldsymbol{d} = \{4, 3, 2, 2, 1\}$ forms the degree sequence of the simple graph



2) $\boldsymbol{d} = \{3, 2, 1\}$ is **not** the degree sequence of any simple graph

3) $\boldsymbol{d} = \{5, 4, 3, 2, 1, 1\}$ is **not** the degree sequence of any simple graph

A sequence of non-negative integers $\boldsymbol{d}$ is called ***graphical*** if there is a *simple* graph G whose degree sequence is $\boldsymbol{d}$.

In this case we say that G ***realizes*** the sequence $\boldsymbol{d}$.

Main questions

Given a sequence of integers $\boldsymbol{d} = \{d_1, \ldots, d_n\}$, $d_1 \geq d_2 \geq \ldots \geq d_n \geq 1$

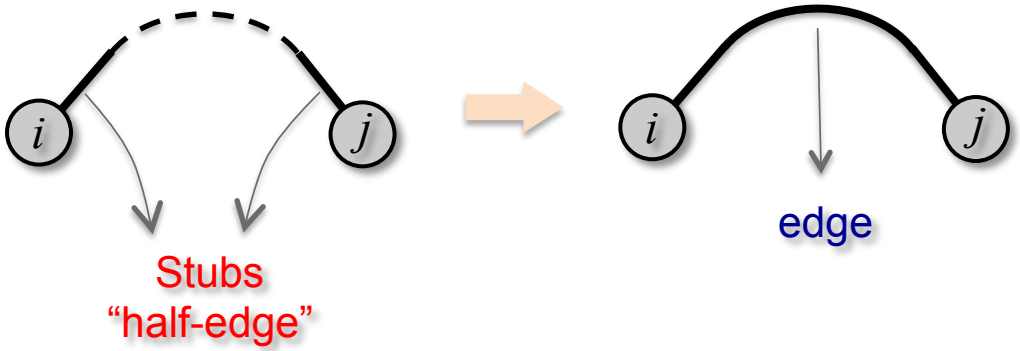
1. Graph Construction:

- How do we *decide* if $\boldsymbol{d}$ is graphical?
- How do we build a simple graph G with sequence $\boldsymbol{d}$ for its degrees?
- How do we build *all possible graphs* with the same degree sequence $\boldsymbol{d}$?

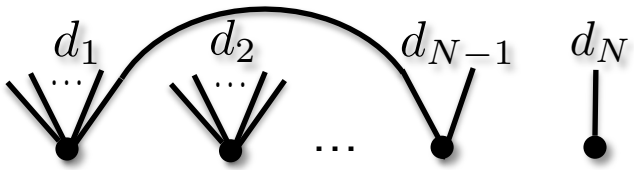
2. Graph Sampling:

- What algorithm would *sample uniformly, or with known weights* from the set of all simple graphs with degree sequence $\boldsymbol{d}$?

Some more jargon:



The sequence $\boldsymbol{d}$ can be represented as a sequence of stubs



“*Realizing a sequence*” means connecting pairs of stubs into edges such that no multiple edges are formed between the same pair of nodes, nor loops, until no stubs are left.

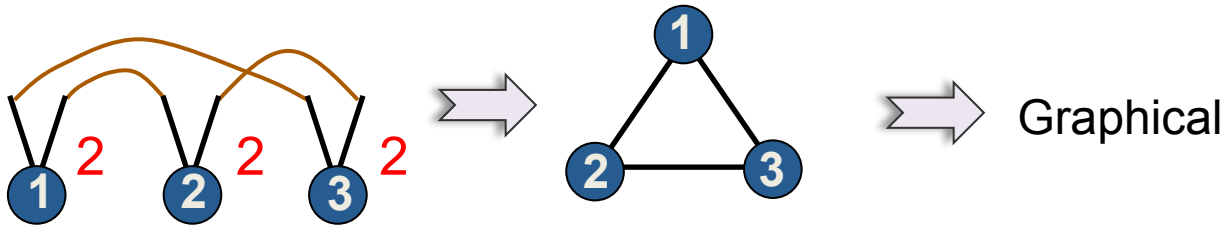
One can also think of *graph construction*, as a process of matching stubs.

Residual degree: the number of unconnected stubs of a node at any given time.

When building graphs, we will always connect all the stubs of the chosen node first before moving on to other nodes with stubs.

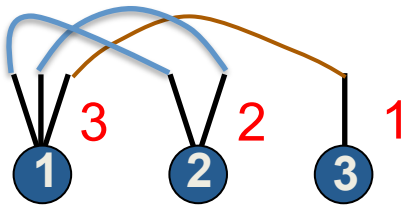
Simple examples:

d ={2,2,2}



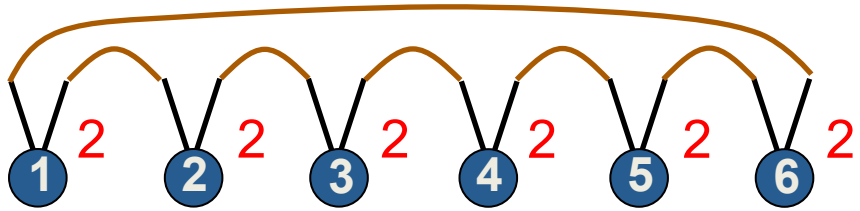
Graphical

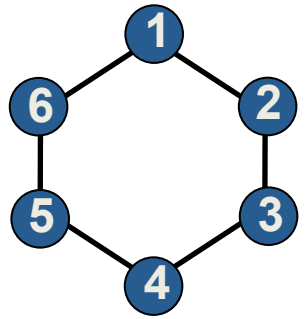
d ={3,2,1}



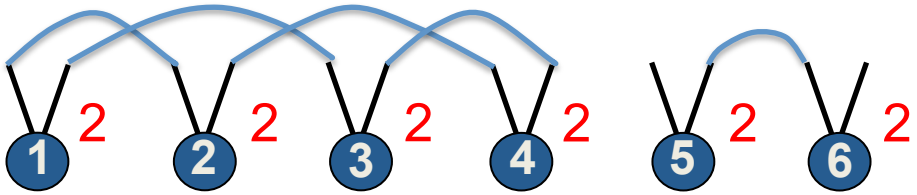
Not graphical

d ={2,2,2,2,2,2}





However, not all connection sequences will result in a simple graph!



Fails to create a simple graph

Characterizations of graphical sequences

Theorem (Erdős, Gallai, EG): *Let* $\boldsymbol{d} = \{d_1, d_2, \ldots, d_N\}$ *be a sequence of non-negative integers with* $d_1 \geq d_2 \geq \ldots \geq d_N \geq 0$ *Then* $\boldsymbol{d}$ *is graphical if and only if (conditions are necessary and sufficient):*

1) $\sum_{i=1}^{N} d_i$ *is even, and*

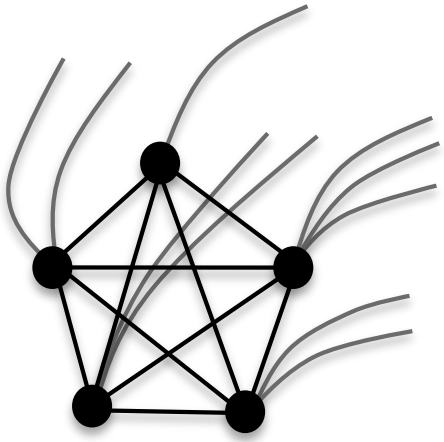
2) $\sum_{i=1}^{k} d_i \leq k(k-1) + \sum_{i=k+1}^{N} \min\{k, d_i\}$ *for all* $1 \leq k \leq N-1$

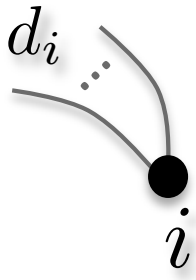
P. Erdős, T. Gallai. Graphs with prescribed degrees of vertices. (Hungarian) *Matematikai Lapok* **11,** 264 (1960).

Note: there can be s

Ex:

$\boldsymbol{d} = \{2, 2, 2, 2, 2, 2,$





The E-G theorem is a

or constructing the graph!

An alternative theore

$k = 5$

sed to build a graph:

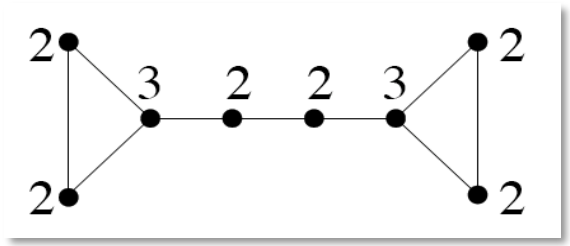
Algorithm (Hakimi, Havel): *Given a graphical sequence, choose a node i (any), and connect **all** its stubs to other nodes with the **largest** residual degrees. Repeat until all stubs are connected into edges.*

V. Havel. A remark on the existence of finite graphs. (Czech) *Časopis Pěst. Mat.* **80** 477 (1955). S.L. Hakimi. On the realizability of a set of integers as degrees of the vertices of a simple graph. *J. SIAM Appl. Math.* **10**, 496 (1962).

This algorithm can be used to characterize graphical sequences: A given sequence of non-negative integers is graphical *iff* the above algorithm finishes in a simple graph.

However, the H-H algorithm cannot construct in general **all** graphs realizing a sequence!

Example: $\boldsymbol{d} = \{3, 3, 2, 2, 2, 2, 2, 2\}$



This is because it restricts the next connection to the node with the largest residual degree.
(so what method can build all realizations?)

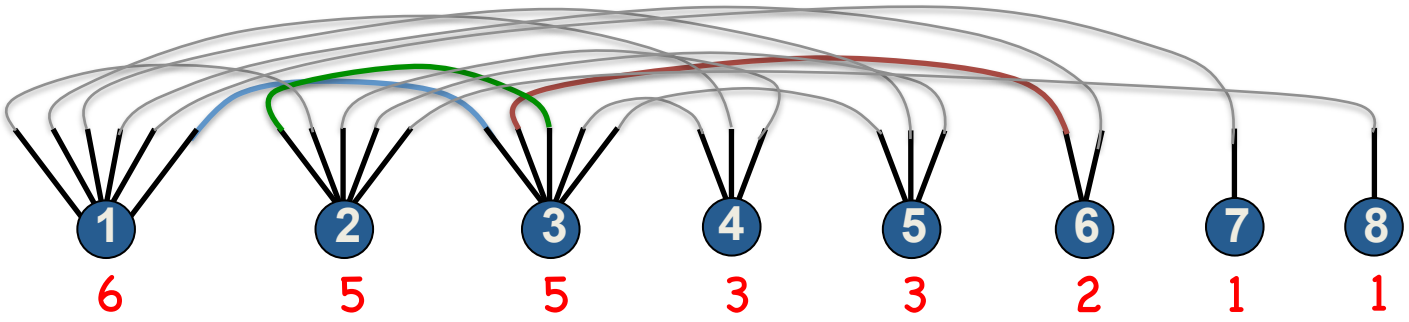
A simple idea:

Connect the stub *arbitrarily*, then use some method to test whether graphicality has been broken by this connection.

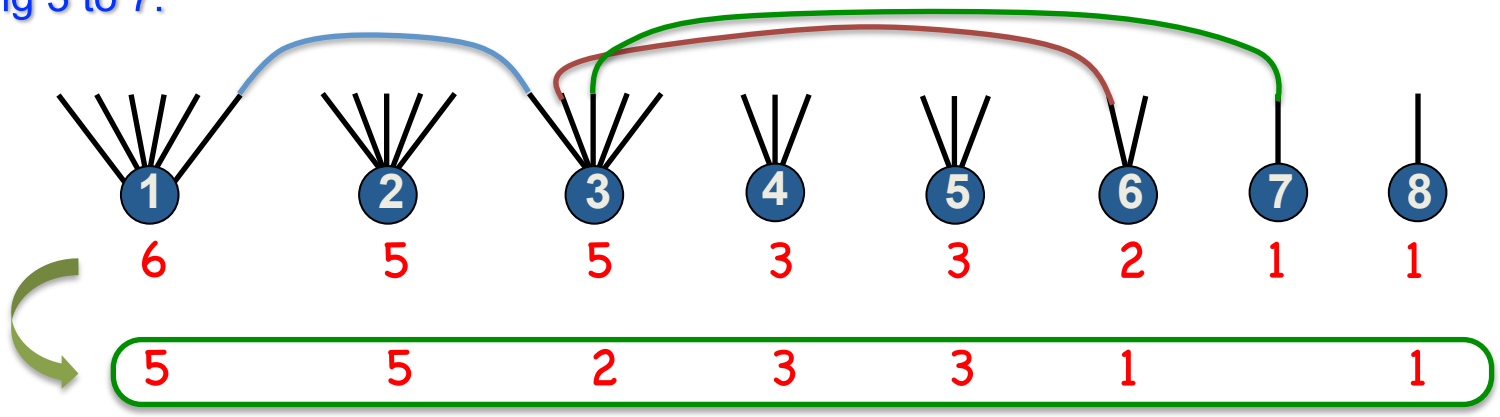
A naïve idea: *use the EG theorem* on the residual sequence after the arbitrary connection has been made… (passing the EG test is necessary anyway).

To see why the E-G theorem is NOT sufficient consider the following example:

$\boldsymbol{d}$ ={6,5,5,3,3,2,1,1} Let us make connections (3,1) and (3,6). Graphicality is still preserved.

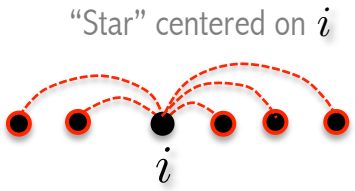


Try connecting 3 to 7:



Passes the EG test! But cannot finish in a simple graph due to constraints.

Star-constrained graphicality



A *star-constraint* on a node i is given by a set of nodes $\mathcal{X}_i$ (forbidden set) to which no connections are allowed from i .

For example, because there are connections already made from i to these nodes.

What we need is a theorem that can tell us whether a given degree sequence is graphical such that connections from a given node i to a set of nodes $\mathcal{X}_i$ are *all avoided*.

Theorem (Star-constrained Graphicality): *Let* $\mathbf{d}: \; d_1 \geq d_2 \geq \ldots \geq d_N \geq 0$ *be a sequence of non-negative integers arranged non-increasingly and* $\mathcal{X}_i$ *be a star-constraint on node* i *with* $|\mathcal{X}_i| \leq N - 1 - d_i$. *Define* $\mathcal{L}_i$ *as the set of the first ("leftmost")* d_i *nodes* ***not from*** $\mathcal{X}_i$. *Then there exist a simple graph realizing* $\mathbf{d}$ *and avoiding connections from* i *to* $\mathcal{X}_i$ *iff the residual sequence* $\mathbf{d'}$ *:*

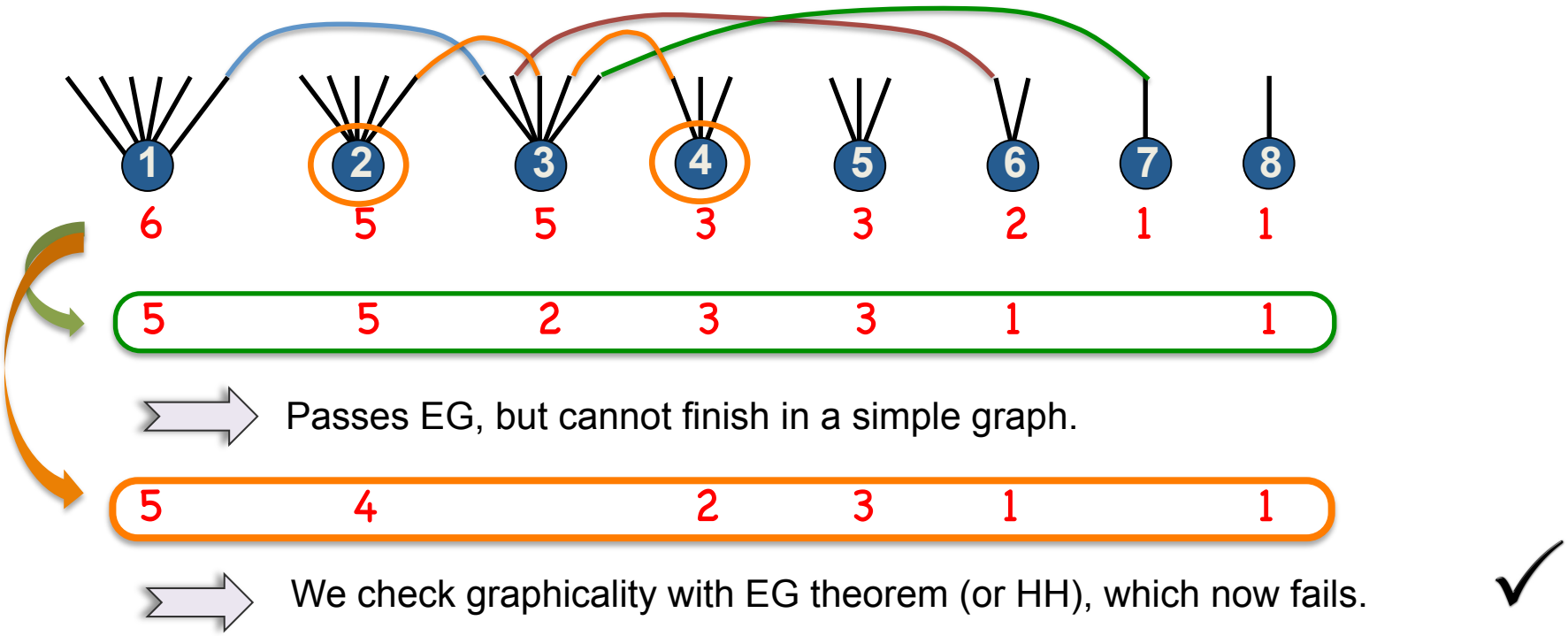
$$d'_j = \begin{cases} d_j - 1 & \text{if } \; j \in \mathcal{L}_i \\ 0 & \text{if } \; j = i \\ d_j & \text{if } \; j \notin \mathcal{L}_i \cup \{i\} \end{cases}$$

is graphical.

H. Kim, Z. Toroczkai, P.L. Erdös, I. Miklós and L.A. Székely. "Degree-based graph construction" *J. Phys. A: Math. Theor.* **42**, 392001 (2009).

Let us see how does this work in our previous example

d ={6,5,5,3,3,2,1,1}



This provides us with the following method that can build all realizations of a graphical sequence:

Procedure:

- *For an arbitrarily chosen node i connect one of its stubs to a stub of another arbitrarily chosen node j only if the residual sequence (after the temporary connection) passes the star-constrained graphicality test.*
- *Repeat with another stub of i until all its stubs are connected away into edges, before moving onto another node.*

Let $\mathcal{G}(\boldsymbol{d})$ denote the set of all simple labeled graphs realizing $\boldsymbol{d}$.

If we specify a systematic way of connecting the stubs, we obtain algorithms that will construct all elements of $\mathcal{G}(\boldsymbol{d})$.

E.g., in:

- H. Kim, Z. Toroczkai, P.L. Erdös, I. Miklós and L.A. Székely. "Degree-based graph construction" *J. Phys. A: Math. Theor.* **42**, 392001 (2009).
- Z. Király. "Recognizing graphic degree sequences and generating all realizations". TR-2011-11, Published by the Egervary Research Group on Combinatorial Optimization, ISSN 1587-4451.

A practical note on implementations:

Sequence graphicality can be decided in linear time (EG) $\mathcal{O}(N)$

- C.I. Del Genio, H. Kim, Z. Toroczkai and K.E. Bassler. "Efficent and exact sampling of simple graphs with given arbitrary degree sequence." PLoS ONE, 5(4) e10012 (2010). - Provides a fast algorithm that does not use multiplications.
- P. Hell, D. Kirkpatrick. "Linear-time certifying algorithms for near-graphical sequences." Discr. Math. **309**, 5703 (2009).

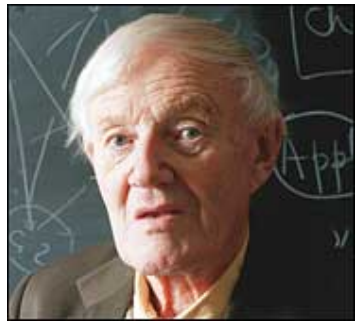
The paper:

- Z. Király. "Recognizing graphic degree sequences and generating all realizations". TR-2011-11, Published by the Egervary Research Group on Combinatorial Optimization, ISSN 1587-4451.

- An $\mathcal{O}(N \log\log N)$ implementation of the Havel-Hakimi algorithm.

- Builds *all* elements from $\mathcal{G}(\boldsymbol{d})$ with complexity $\mathcal{O}(N^2)$ between realizations.

A note on the larger context:


William T. Tutte, 1918-2002.

Given a simple graph $G(V,E)$ and a function: $f : V(G) \to \mathbb{N} \cup \{0\}$

an *f-factor* of G is a subgraph H such that:

$$d_H(v) = f(v)\ , \quad \text{for all} \ \ v \in V$$

(A *1-factor* is a matching.) W.T. Tutte. "The factors of graphs." *Canad. J. Math.* **4**, 314 (1952).

When $G = K_N$ the f-factor problem is nothing but the degree-based graph construction problem !

We provided a greedy algorithm to construct f-factors in: $K_N \setminus S_k$ where S_k is a star graph of k nodes centered on some (arbitrary) node.

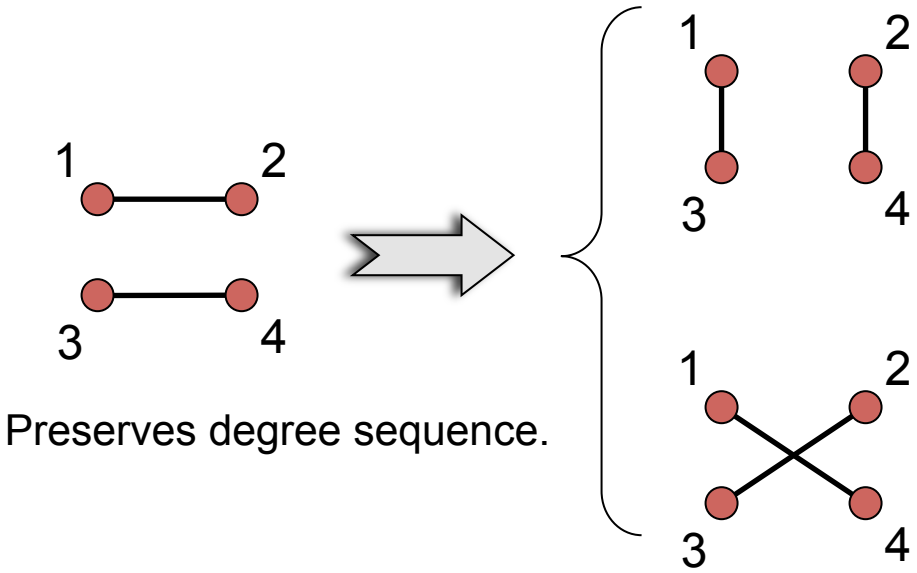
The Sampling Problem

$|\mathcal{G}(\boldsymbol{d})|$ is typically enormous, cannot build all realizations, one needs to sample from $\mathcal{G}(\boldsymbol{d})$

Ideally, for modeling purposes we'd like to sample *uniformly at random*.

Two main approaches to graph sampling:

1. **M**arkov **C**hain **M**onte **C**arlo (MCMC): Uses edge swaps (switches). Catherine, Peter



Preserves degree sequence.

Problem: mixing time is not known in general.

Ryser:

If G_1 and G_2 are two simple graphs with the same degree sequence, then a sequence of edge swaps transforms one into another.

H.J. Ryser . "Combinatorial properties of matrices of zeros and ones" *Canad. J. Math.* **9** 371 (1957).

R. A. Brualdi "Matrices of zeros and ones with fixed row and column sum vectors" *Lin. Alg. Appl.* **33**, 159 (1980).

R. Taylor "Constrained switchings in graphs" *SIAM J. ALG. DISC. METH.* **3,** 115 (1982)

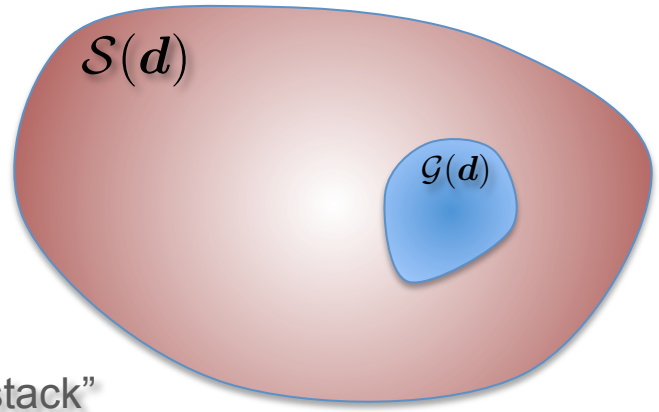
2. Direct Construction: - Matching stubs

Configuration Model (CM):
- Connect a pair of stubs uniformly at random.
- If multiple (parallel) edges or self-loops, reject then restart.

B. Bollobas. *Eur. J Comb.* **1**:311-316 (1980) E. Bender, E. Canfield *J. Comb. Th. A* **24** 296-307 (1978)
M Molloy, B. Reed. *Rand. Struct. Alg.* **6**:161-179 (1995)

Advantage: uniform sampler.

Disadvantage:
- Can have many rejections depending on $\boldsymbol{d}$.



$\mathcal{S}(\boldsymbol{d})$: set of all graphs with degree seq $\boldsymbol{d}$

$\mathcal{G}(\boldsymbol{d})$: set of all *simple* graphs with degree seq $\boldsymbol{d}$

Worst case: "a needle in the haystack"

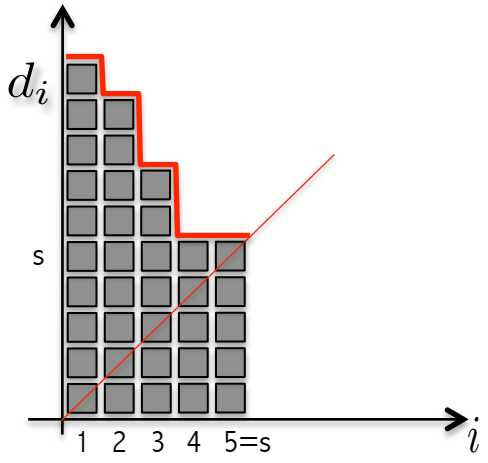
For those who favor tails of distributions:

H-sequences

Define the ***large degree tail*** of a sequence $\boldsymbol{d}$ of non-negative integers the subsequence:

$\{d_1, d_2, \ldots, d_s\}$ with

$d_1 \geq d_2 \geq \ldots \geq d_s \geq s\,, \qquad d_{s+1} < s+1$



Pick your favorite "tail sequence" of integers: $h_1 \geq h_2 \geq \ldots \geq h_s \geq s \geq 1$.

Assuming we really don't care much about the low degree part, we will extend it further such that the full sequence is graphical and it has some other properties.

In particular, let us introduce the sequence $\bar{h}_i = \sum_{j=1}^{s} \theta(h_j + 1 - s - i)$, $1 \leq i \leq h_1 + 1 - s$.

$\theta(n) = \begin{cases} 1 , & n \geq 0 \\ 0 , & n < 0 \end{cases}$

Theorem (H-sequence): *For an arbitrary sequence of non-negative integers* $h_1 \geq \ldots \geq h_s \geq s \geq 1$ *the sequence* $H_1 \geq \ldots \geq H_n$ *defined by:*

$$H_i = \begin{cases} h_i , & 1 \leq i \leq s , \\ \bar{h}_{i-s} , & s+1 \leq i \leq n = h_1 + 1 \end{cases}$$

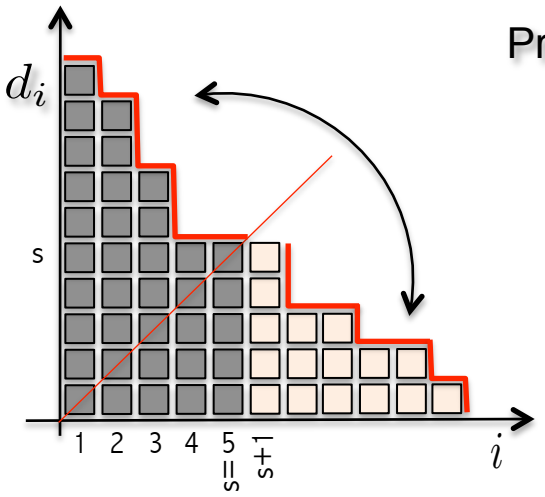
is graphical.

Moreover, we have for all $1 \leq k \leq s$:

$$\sum_{i=1}^{k} H_i = k(k-1) + \sum_{i=k+1}^{n} \min\{k, H_i\} .$$

(that is, we have equalities in the EG test)

In words: the low-degree part is obtained by mirroring the large-degree tail onto the first bisector, shifted by unity:

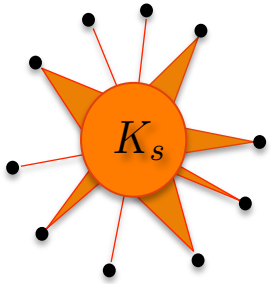


Properties:

- has <u>exactly one</u> graphical realization by a simple graph
- the first s nodes are connected into a complete graph K_s
- there are no connections between nodes in $\{s+1,\ldots,n\}$
- among all simple graphs with the <u>same tail</u> $\boldsymbol{h}$ the H-sequence graph is the "tightest", i.e., has the smallest volume, given by:

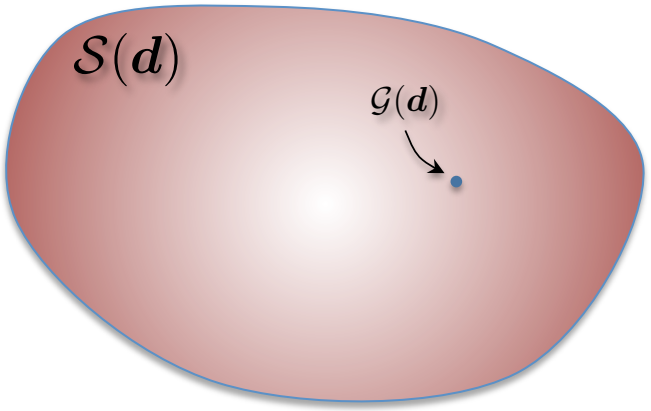
$$\sum_{i=1}^{n} H_i = 2\sum_{i=1}^{s} h_i - s(s-1)$$

- its diameter is 2



The "Sun Graph"

The chances of hitting the Sun Graph by the CM algorithm is astronomically small for large $n = h_1 + 1$



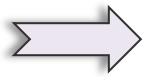
M. Koren. J. Comb. Theory. B **21**, 235 (1976).

Can we do better than the CM algorithm?

The graph sampling algorithm

Given a graphical sequence $d_1 \geq \ldots \geq d_n \geq 1$

1. Choose the first node in the sequence as the "work" node
2. Build the set of allowed nodes, $\mathcal{A}$ that can be connected to the work node
3. Choose uniformly at random a node a in $\mathcal{A}$ and connect it to the work node
   - 3.1 If a has still stubs, add it to the set of forbidden nodes.
   - 3.2 Otherwise, remove it from residual sequence
4. Repeat from 2 to 3.2 until all stubs of the work node are connected away.
5. Remove the work node from the sequence
6. Repeat the whole procedure until we end with a simple graph.



Biased sampling, but it provides the weight of the sample.

C.I. Del Genio et.al. *PLoS ONE* **5**(4), e10012 , (2010).

Measuring network observables uniformly

$$\langle Q \rangle = \frac{\sum_{i=1}^{K} \omega_{S_i} Q(S_i)}{\sum_{i=1}^{K} \omega_{S_i}}$$

K : number of samples

$\langle Q \rangle$: average of the observable

$Q(S_i)$: observable measured for sample S_i

$\omega(S_i)$: inverse of the relative probability probability for S_i to occur

$$\Longrightarrow \quad \omega = \prod_{i=1}^{m} \frac{1}{\bar{d}_i!} \prod_{j=1}^{\bar{d}_i} k_{i_j}$$

k_{i_j} : Size of allowed set

$\bar{d}_i$: residual degree of work node

m : number of work nodes ($\leq N$-1)

C.I. Del Genio, H. Kim, Z. Toroczkai & K.E. Bassler. *PLoS ONE*, **5**(4), e10012 , (2010).

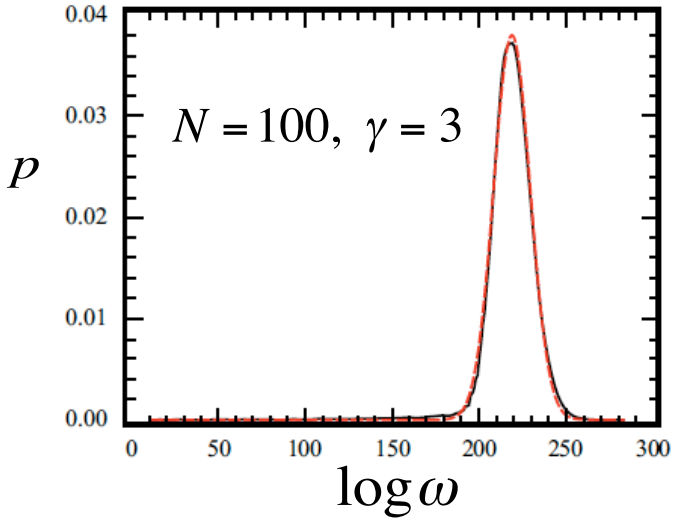
H. Kim, C.I. Del Genio, K.E. Bassler & Z. Toroczkai.. *New J. Phys.* **14**, 02312 (2012).

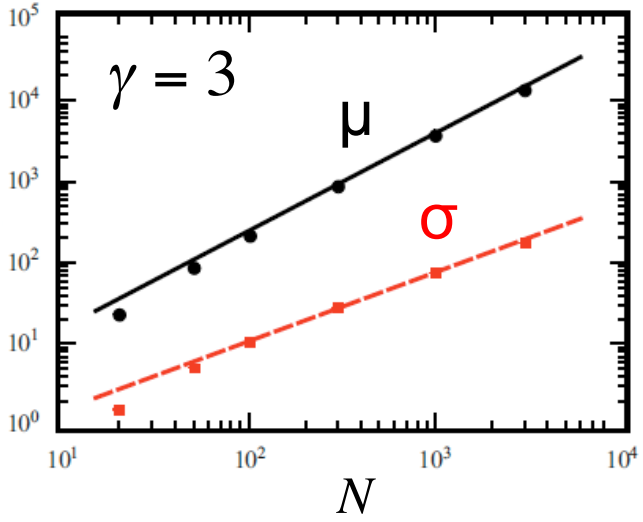
$$\ln \omega = \sum_{i=1}^{m} \left[\left(\sum_{j=1}^{\bar{d}_i} \ln k_{i_j} \right) - \ln \left(\bar{d}_i \right) \right]$$

Many realizations ⇨ log-normal distribution

Sample weights for an ensemble of ***power-law sequences*** chosen from $P(d) \sim d^{-\gamma}$

Simulations: 2×10^4 graphical sequences, 10^6 samples for each, for a total of 2×10^{10} samples.





- Upper bound for worst case complexity $\mathcal{O}(NM)$

If number of edges $M = \mathcal{O}(N)$ ➔ Complexity $\mathcal{O}(N^2)$

If number of edges $M = \mathcal{O}(N^2)$ ➔ Complexity $\mathcal{O}(N^3)$

Sampling summary

• Provided an algorithm does not use swaps, instead it is based on the star-constrained graph construction results.

• Unlike the Configuration Model based method, it is rejection free: it always ends in a simple graph realization of the degree sequence.

From Charo: "The CM model failed to produce even a single sample of a sequence from $P(d) = const.$ with $N = 100$ after running for more than **24** hours, while our algorithm produced 10^4 realizations of the very same sequence in 30 seconds."

• It produces statistically independent samples.

• It provides the sample weight, which in turn can be used to compute network observables as if sampled uniformly or by some preferred distribution.

• Drawback: due to the log-normal distribution, one has to draw many samples.

Directed graph construction and sampling

All of the above can be extended to bi-degree sequences (bds) to realize directed simple graphs with the same bds:

$$\boldsymbol{D} = (\boldsymbol{d}^{(i)}, \boldsymbol{d}^{(o)}) = \{(d_1^{(i)}, d_1^{(o)}), \ldots (d_n^{(i)}, d_n^{(o)})\}$$

EG $\longmapsto$ Fulkerson-Ryser (fast implementation: Charo and Kevin)

HH $\longmapsto$ P.L. Erdős, I. Miklós & Z. Toroczkai. *Electron. J. Comb*. **17**(1)**,** R66, (2010).
D.J. Kleitman & D.L. Wang. *Discr. Math.* **6**, 79 (1973).

Sampling: H. Kim, C.I. Del Genio, K.E. Bassler & Z. Toroczkai. *New J. Phys.* **14**, 02312 (2012).