Combinatorial Analysis of Growth Models for Series-Parallel Networks

Markus Kuba and Alois Panholzer

Technikum Wien and TU Wien, Austria

AofA 2016, Krakow, 07.07.2016

Series-parallel networks

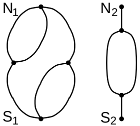
Series-parallel networks (SP-nets):

- two-terminal graphs: distinguished vertices source and sink
- generated recursively by two composition operations:
    - **parallel composition:**
      sources and sinks of two SP-nets are merged
    - **series composition:**
      sink of one SP-net merged with source of other SP-net

Series-parallel networks

Series-parallel networks (SP-nets):

- two-terminal graphs: distinguished vertices source and sink
- generated recursively by two composition operations:
    - **parallel composition:**
      sources and sinks of two SP-nets are merged

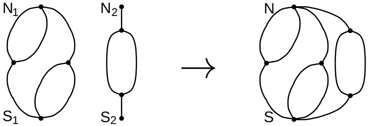


    - **series composition:**
      sink of one SP-net merged with source of other SP-net

Series-parallel networks

Series-parallel networks (SP-nets):

- two-terminal graphs: distinguished vertices source and sink
- generated recursively by two composition operations:
  - **parallel composition:**
    sources and sinks of two SP-nets are merged

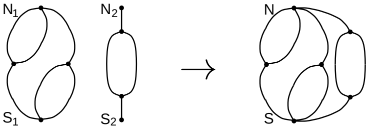


  - **series composition:**
    sink of one SP-net merged with source of other SP-net

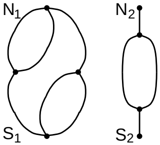
Series-parallel networks

Series-parallel networks (SP-nets):

- two-terminal graphs: distinguished vertices source and sink
- generated recursively by two composition operations:
  - **parallel composition:**
    sources and sinks of two SP-nets are merged



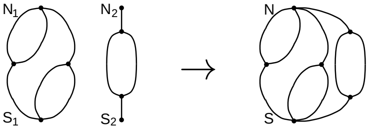
  - **series composition:**
    sink of one SP-net merged with source of other SP-net



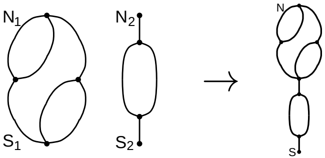
Series-parallel networks

Series-parallel networks (SP-nets):

- two-terminal graphs: distinguished vertices source and sink
- generated recursively by two composition operations:
  - **parallel composition:**
    sources and sinks of two SP-nets are merged



  - **series composition:**
    sink of one SP-net merged with source of other SP-net



Applications of SP-nets:

- model flow in bipolar networks (electric circuits, etc.)
- computational complexity theory: some in general NP-complete graph problems are solvable in linear time for SP-nets (max. independent set, . . .)

Models for SP-nets/SP-graphs:

various studies of typical behaviour of structural quantities in SP-nets/SP-graphs

- **uniform models:** all SP-nets/SP-graphs of given size are equally likely
  e.g., Bodirsky, Gimenez, Kang, Noy [2007]; Bernasconi, Panagiotou, Steger [2008]; Drmota, Gimenez, Noy [2010,2011]; Hofri, Li, Mahmoud [2016]
- **stochastic growth models:** introduced by Hosam Mahmoud [2013, 2014]

Series-parallel networks

Applications of SP-nets:

- model flow in bipolar networks (electric circuits, etc.)
- computational complexity theory: some in general NP-complete graph problems are solvable in linear time for SP-nets (max. independent set, ...)

Models for SP-nets/SP-graphs:

various studies of typical behaviour of structural quantities in SP-nets/SP-graphs

- **uniform models:** all SP-nets/SP-graphs of given size are equally likely
  e.g., Bodirsky, Gimenez, Kang, Noy [2007]; Bernasconi, Panagiotou, Steger [2008]; Drmota, Gimenez, Noy [2010,2011]; Hofri, Li, Mahmoud [2016]
- **stochastic growth models:** introduced by Hosam Mahmoud [2013, 2014]

Series-parallel networks

Applications of SP-nets:

- model flow in bipolar networks (electric circuits, etc.)
- computational complexity theory: some in general NP-complete graph problems are solvable in linear time for SP-nets (max. independent set, ...)

Models for SP-nets/SP-graphs:

various studies of typical behaviour of structural quantities in SP-nets/SP-graphs

- **uniform models:** all SP-nets/SP-graphs of given size are equally likely
  e.g., Bodirsky, Gimenez, Kang, Noy [2007]; Bernasconi, Panagiotou, Steger [2008]; Drmota, Gimenez, Noy [2010,2011]; Hofri, Li, Mahmoud [2016]
- **stochastic growth models:** introduced by Hosam Mahmoud [2013, 2014]

Series-parallel networks

Applications of SP-nets:

- model flow in bipolar networks (electric circuits, etc.)
- computational complexity theory: some in general NP-complete graph problems are solvable in linear time for SP-nets (max. independent set, . . .)

Models for SP-nets/SP-graphs:

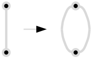
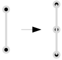
various studies of typical behaviour of structural quantities in SP-nets/SP-graphs

- **uniform models:** all SP-nets/SP-graphs of given size are equally likely
  e.g., Bodirsky, Gimenez, Kang, Noy [2007]; Bernasconi, Panagiotou, Steger [2008]; Drmota, Gimenez, Noy [2010,2011]; Hofri, Li, Mahmoud [2016]
- **stochastic growth models:** introduced by Hosam Mahmoud [2013, 2014]

Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

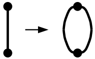
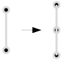
- parallel doubling: 
- serial doubling: 

Example:

Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

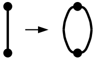
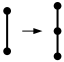
- parallel doubling: 
- serial doubling: 

Example:

Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

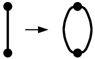
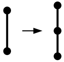
- parallel doubling: 
- serial doubling: 

Example:

Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

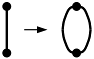
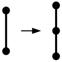
- parallel doubling: 
- serial doubling: 

Example:

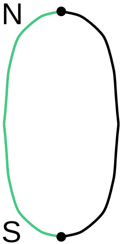
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

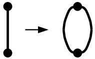
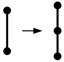
Example:



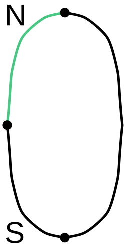
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

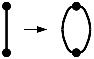
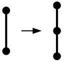
Example:



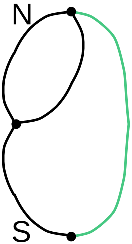
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

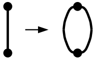
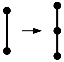
Example:



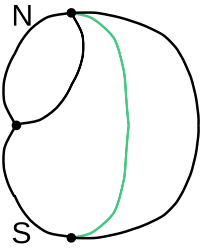
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

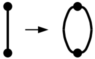
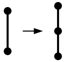
Example:



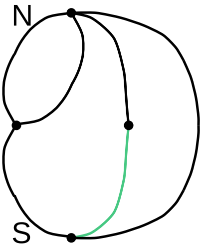
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

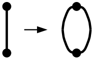
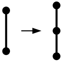
Example:



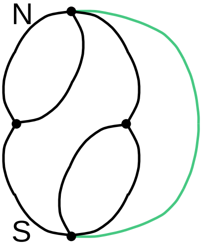
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

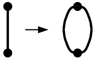
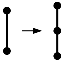
Example:



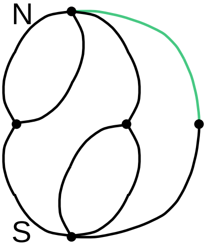
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

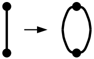
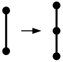
Example:



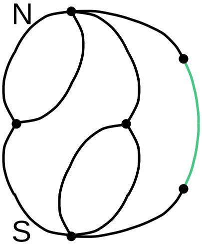
Series-parallel networks

Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

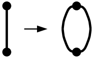
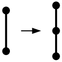
- parallel doubling: 
- serial doubling: 

Example:

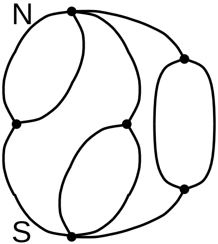


Generating SP-nets by edge-duplications:

each SP-net can be generated by basic edge-duplication rules:

- parallel doubling: 
- serial doubling: 

Example:



Stochastic growth models for SP-nets

Bernoulli model [Mahmoud, 2013]:

- **step 1:** start with single edge
- **step n:** select edge $e =$ uniformly at random
  - with probability p: parallel duplication of e

  - with probability $q = 1 - p$: serial duplication of e

Stochastic growth models for SP-nets

Bernoulli model [Mahmoud, 2013]:

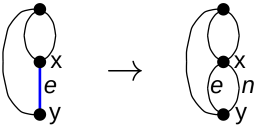
- **step 1:** start with single edge 1
- **step n:** select edge $e =$ x–y uniformly at random
  - with probability p: parallel duplication of e

  - with probability $q = 1 - p$: serial duplication of e

Stochastic growth models for SP-nets

Bernoulli model [Mahmoud, 2013]:

- **step 1:** start with single edge
- **step n:** select edge $e =$ uniformly at random
  - with probability p: parallel duplication of e



  - with probability $q = 1 - p$: serial duplication of e

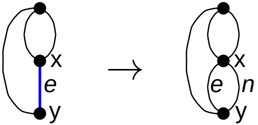
Stochastic growth models for SP-nets

Bernoulli model [Mahmoud, 2013]:

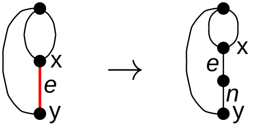
- **step 1:** start with single edge 

- **step n:** select edge $e =$ 

 uniformly at random
  - with probability p: parallel duplication of e



  - with probability $q = 1 - p$: serial duplication of e



Stochastic growth models for SP-nets

Binary model [Mahmoud, 2014]:

- **step 1:** start with single edge 

- **step** n**:** select edge $e =$  uniformly at random

  - if out-degree $d^+(x) = 1$: parallel duplication of e

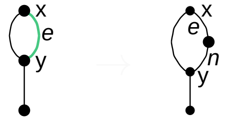
  - if out-degree $d^+(x) = 2$: serial duplication of e

Stochastic growth models for SP-nets

Binary model [Mahmoud, 2014]:

- **step 1:** start with single edge
- **step n:** select edge $e =$ uniformly at random
  - if out-degree $d^+(x) = 1$: parallel duplication of e

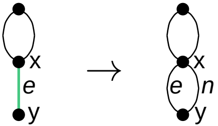
  - if out-degree $d^+(x) = 2$: serial duplication of e



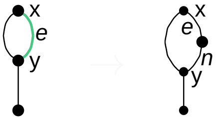
Stochastic growth models for SP-nets

Binary model [Mahmoud, 2014]:

- **step 1:** start with single edge 1
- **step n:** select edge $e =$ x y uniformly at random
  - if out-degree $d^+(x) = 1$: parallel duplication of e



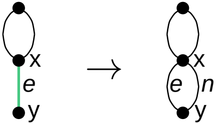
  - if out-degree $d^+(x) = 2$: serial duplication of e



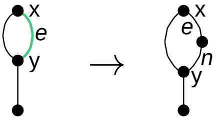
Stochastic growth models for SP-nets

Binary model [Mahmoud, 2014]:

- **step** 1**:** start with single edge 
- **step** n**:** select edge $e =$  uniformly at random
  - if out-degree $d^+(x) = 1$: parallel duplication of e



  - if out-degree $d^+(x) = 2$: serial duplication of e



Combinatorial description

Stochastic growth rule

→ "bottom-up" analysis of quantities in SP-net models via Pólya-Eggenberger urn models [Mahmoud, 2013 & 2014]: nodes of small degree, degree of the source, length of random source-to-sink path

Combinatorial description of SP-net models

→ capture growth process via increasing tree models

→ allows combinatorial decomposition of structure

→ "top-down" approach for analysis of quantities in SP-nets by considering parameters in corresponding tree models

→ approach can be extended to further growth rules for SP-nets

Combinatorial description

Stochastic growth rule

→ "bottom-up" analysis of quantities in SP-net models via Pólya-Eggenberger urn models [Mahmoud, 2013 & 2014]: nodes of small degree, degree of the source, length of random source-to-sink path

Combinatorial description of SP-net models

→ capture growth process via increasing tree models

→ allows combinatorial decomposition of structure

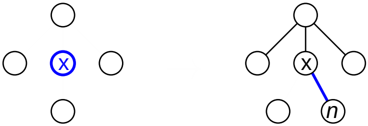
→ "top-down" approach for analysis of quantities in SP-nets by considering parameters in corresponding tree models

→ approach can be extended to further growth rules for SP-nets

Combinatorial description of SP-net models

Bernoulli model → edge-coloured recursive trees:

- **step** 1**:** start with single node ①
- **step** n**:** select node ⓧ uniformly at random
  - with probability p: attach node n to x with blue edge

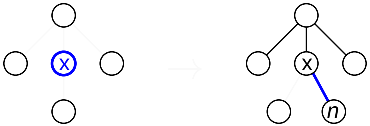


  - with probability $q = 1 - p$: attach node n to x with red edge

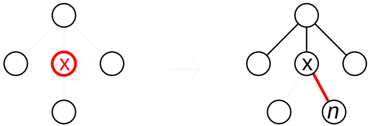
Combinatorial description of SP-net models

Bernoulli model $\rightarrow$ edge-coloured recursive trees:

- **step** 1: start with single node ①
- **step** n: select node ⓧ uniformly at random
  - with probability p: attach node n to x with blue edge



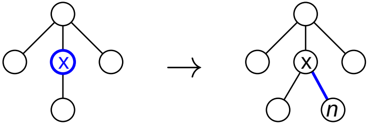
  - with probability $q = 1 - p$: attach node n to x with red edge



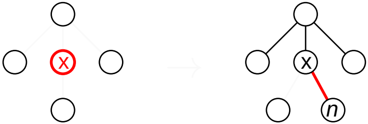
Combinatorial description of SP-net models

Bernoulli model $\rightarrow$ edge-coloured recursive trees:

- **step 1:** start with single node ①
- **step n:** select node ⓧ uniformly at random
  - with probability p: attach node n to x with blue edge



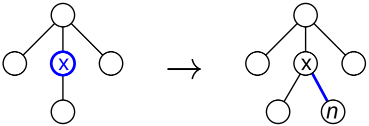
  - with probability $q = 1 - p$: attach node n to x with red edge



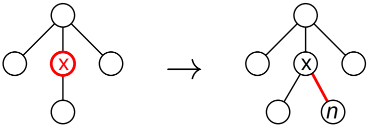
Combinatorial description of SP-net models

Bernoulli model $\rightarrow$ edge-coloured recursive trees:

- **step** 1**:** start with single node ①
- **step** n**:** select node ⓧ uniformly at random
  - with probability p: attach node n to x with blue edge



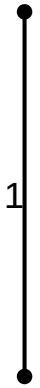
  - with probability $q = 1 - p$: attach node n to x with red edge



Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree

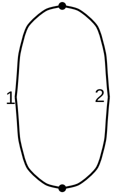
- edge-coloured unordered increasing trees
- parallel duplication $\hat{=}$ blue edge
- serial duplication $\hat{=}$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

$$\boxed{\mathcal{T} \;=\; ① \;\dot\cup\; ① \times \textsc{Set}(\{\,|\,\} \times \mathcal{T} \;\dot\cup\; \{\,|\,\} \times \mathcal{T})}$$

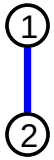
Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree



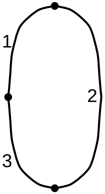
- edge-coloured unordered increasing trees
- parallel duplication $\hat{=}$ blue edge
- serial duplication $\hat{=}$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

$$\boxed{\mathcal{T} = ① \,\dot{\cup}\, ① \times \textsc{Set}(\{\,|\,\} \times \mathcal{T} \,\dot{\cup}\, \{\,|\,\} \times \mathcal{T})}$$

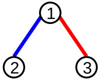
Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree



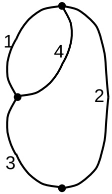
- edge-coloured unordered increasing trees
- parallel duplication $\triangleq$ blue edge
- serial duplication $\triangleq$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

$$\boxed{\mathcal{T} = 1 \ \dot{\cup} \ 1 \times \textsc{Set}(\{ \,|\, \} \times \mathcal{T} \ \dot{\cup} \ \{ \,|\, \} \times \mathcal{T})}$$

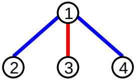
Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree

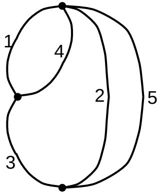


- edge-coloured unordered increasing trees
- parallel duplication $\hat{=}$ blue edge
- serial duplication $\hat{=}$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

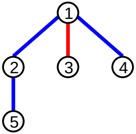
$$\boxed{\mathcal{T} \;=\; ① \;\dot{\cup}\; ① \times \textsc{Set}\big(\{\,|\,\} \times \mathcal{T} \;\dot{\cup}\; \{\,|\,\} \times \mathcal{T}\big)}$$

Example:

SP-net



edge-coloured recursive tree

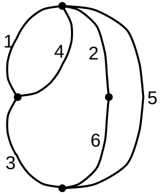


- edge-coloured unordered increasing trees
- parallel duplication $\hat{=}$ blue edge
- serial duplication $\hat{=}$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

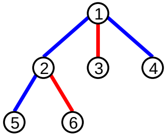
$$\boxed{\mathcal{T} \;=\; \textcircled{1} \;\dot\cup\; \textcircled{1} \times \textsc{Set}\big(\{\,\color{blue}{|}\,\} \times \mathcal{T} \;\dot\cup\; \{\,\color{red}{|}\,\} \times \mathcal{T}\big)}$$

Example:

SP-net



edge-coloured recursive tree



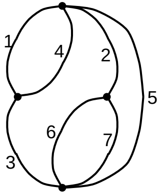
- edge-coloured unordered increasing trees
- parallel duplication $\hat{=}$ blue edge
- serial duplication $\hat{=}$ red edge
- “top-down” decomposition of tree family $\mathcal{T}$:

$$\boxed{\mathcal{T} \;=\; ① \;\dot{\cup}\; ① \times \textsc{Set}\big(\{\,|\,\} \times \mathcal{T} \;\dot{\cup}\; \{\,|\,\} \times \mathcal{T}\big)}$$

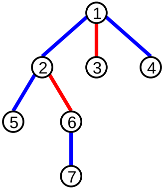
Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree



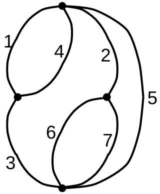
- edge-coloured unordered increasing trees
- parallel duplication ≙ blue edge
- serial duplication ≙ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

$$\mathcal{T} = \textcircled{1} \;\dot\cup\; \textcircled{1} \times \textsc{Set}\big(\{\,|\,\} \times \mathcal{T} \;\dot\cup\; \{\,|\,\} \times \mathcal{T}\big)$$

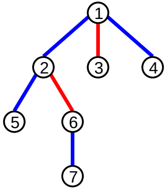
Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree



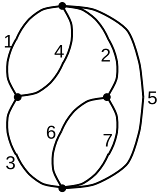
- edge-coloured unordered increasing trees
- parallel duplication $\widehat{=}$ blue edge
- serial duplication $\widehat{=}$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

$$\mathcal{T} = \textcircled{1} \;\dot\cup\; \textcircled{1} \times \textsc{Set}\left(\{\,|\,\} \times \mathcal{T} \;\dot\cup\; \{\,|\,\} \times \mathcal{T}\right)$$

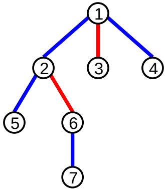
Combinatorial description of SP-net models

Example:

SP-net



edge-coloured recursive tree



- edge-coloured unordered increasing trees
- parallel duplication $\hat{=}$ blue edge
- serial duplication $\hat{=}$ red edge
- "top-down" decomposition of tree family $\mathcal{T}$:

$$\boxed{\mathcal{T} \;=\; \textcircled{1} \;\dot{\cup}\; \textcircled{1} \times \textsc{Set}\big(\{\,|\,\} \times \mathcal{T} \;\dot{\cup}\; \{\,|\,\} \times \mathcal{T}\big)}$$

Combinatorial description of SP-net models

Binary model $\rightarrow$ bucket recursive trees:

- nodes in tree can hold up to two labels
- **step** 1: start with single node containing label ①
- **step** n: select label $1 \leq j < n$ uniformly at random
  - if node x containing j is not saturated: label n will be inserted into x

  - if node x containing j is saturated: new node containing label n will be attached to x

Combinatorial description of SP-net models

Binary model $\rightarrow$ bucket recursive trees:

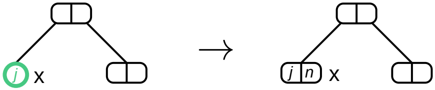
- nodes in tree can hold up to two labels
- **step** 1**:** start with single node containing label ①
- **step** n**:** select label $1 \leq j < n$ uniformly at random
    - if node x containing j is not saturated:
      label n will be inserted into x

    - if node x containing j is saturated:
      new node containing label n will be attached to x

Combinatorial description of SP-net models

Binary model $\rightarrow$ bucket recursive trees:

- nodes in tree can hold up to two labels
- **step** 1**:** start with single node containing label ①
- **step** n**:** select label $1 \leq j < n$ uniformly at random
  - if node x containing j is not saturated:
    label n will be inserted into x

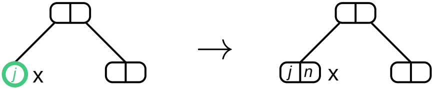


  - if node x containing j is saturated:
    new node containing label n will be attached to x

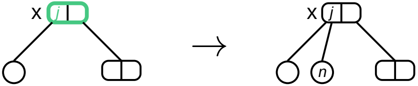
Combinatorial description of SP-net models

Binary model → bucket recursive trees:

- nodes in tree can hold up to two labels
- **step** 1: start with single node containing label ①
- **step** n: select label $1 \leq j < n$ uniformly at random
  - if node x containing j is not saturated: label n will be inserted into x



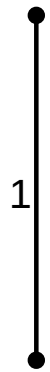
  - if node x containing j is saturated: new node containing label n will be attached to x



Combinatorial description of SP-net models

Example:

SP-net

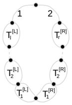


bucket recursive tree

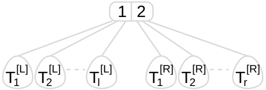
- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} = (1) \,\dot{\cup}\, \boxed{1\,|\,2} \times \mathrm{SET}(\mathcal{B}) \times \mathrm{SET}(\mathcal{B})}$$

- tree decomposition reflects subblock-structure



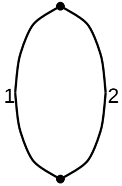
$\longleftrightarrow$



Combinatorial description of SP-net models

Example:

SP-net



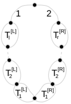
bucket recursive tree

$\boxed{1 \mid 2}$

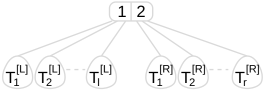
- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} \;=\; \textcircled{1} \;\dot{\cup}\; \boxed{1 \mid 2} \times \textsc{Set}(\mathcal{B}) \times \textsc{Set}(\mathcal{B})}$$

- tree decomposition reflects subblock-structure



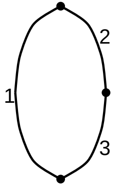
$\longleftrightarrow$



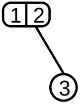
Combinatorial description of SP-net models

Example:

SP-net



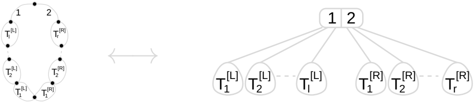
bucket recursive tree



- “top-down” decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} = 1 \;\dot{\cup}\; \boxed{1\,|\,2} \times \mathrm{SET}(\mathcal{B}) \times \mathrm{SET}(\mathcal{B})}$$

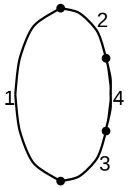
- tree decomposition reflects subblock-structure



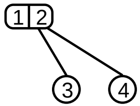
Combinatorial description of SP-net models

Example:

SP-net



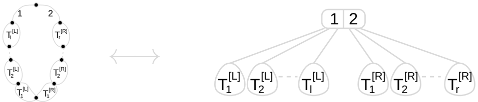
bucket recursive tree



- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} = \text{①} \,\dot{\cup}\, \boxed{1|2} \times \textsc{Set}(\mathcal{B}) \times \textsc{Set}(\mathcal{B})}$$

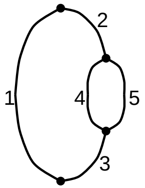
- tree decomposition reflects subblock-structure



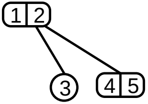
Combinatorial description of SP-net models

Example:

SP-net



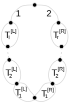
bucket recursive tree



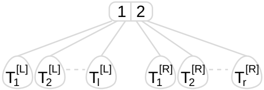
- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} \;=\; 1 \;\dot{\cup}\; \boxed{1\,|\,2} \times \textsc{Set}(\mathcal{B}) \times \textsc{Set}(\mathcal{B})}$$

- tree decomposition reflects subblock-structure



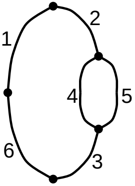
$\longleftrightarrow$



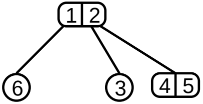
Combinatorial description of SP-net models

Example:

SP-net



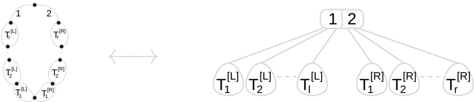
bucket recursive tree



- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

$$\mathcal{B} = \textcircled{1} \;\dot{\cup}\; \boxed{1\,|\,2} \times \textsc{Set}(\mathcal{B}) \times \textsc{Set}(\mathcal{B})$$

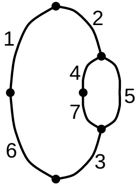
- tree decomposition reflects subblock-structure



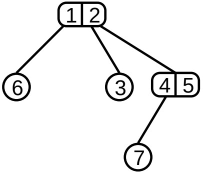
Combinatorial description of SP-net models

Example:

SP-net



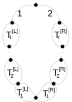
bucket recursive tree



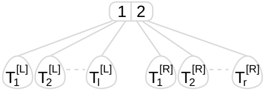
- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} = \text{①} \;\dot{\cup}\; \boxed{1\,|\,2} \times \textsc{Set}(\mathcal{B}) \times \textsc{Set}(\mathcal{B})}$$

- tree decomposition reflects subblock-structure



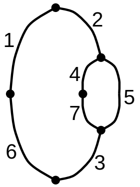
$\longleftrightarrow$



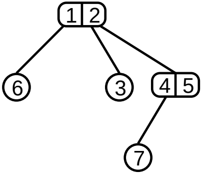
Combinatorial description of SP-net models

Example:

SP-net



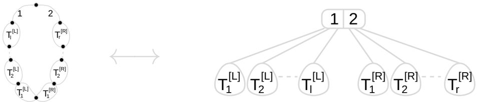
bucket recursive tree



- “top-down” decomposition of bucket rec. tree family $\mathcal{B}$:

$$\boxed{\mathcal{B} \;=\; \textcircled{1} \;\dot{\cup}\; \boxed{1|2} \times \mathrm{SET}(\mathcal{B}) \times \mathrm{SET}(\mathcal{B})}$$

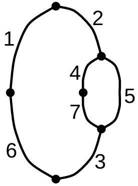
- tree decomposition reflects subblock-structure



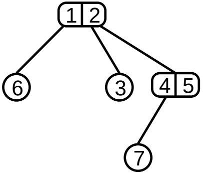
Combinatorial description of SP-net models

Example:

SP-net



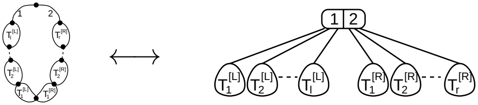
bucket recursive tree



- "top-down" decomposition of bucket rec. tree family $\mathcal{B}$:

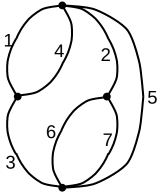
$$\boxed{\mathcal{B} \;=\; \textcircled{1} \;\dot{\cup}\; \boxed{1|2} \times \mathrm{SET}(\mathcal{B}) \times \mathrm{SET}(\mathcal{B})}$$

- tree decomposition reflects subblock-structure

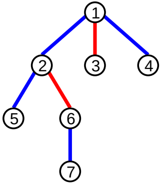


Parameter I: source degree for Bernoulli model

SP-net



edge-coloured recursive tree

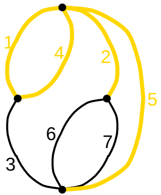


Generating functions approach:

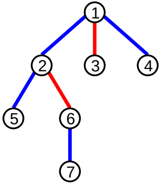
- count $\sharp$ of trees of order n with k blue edges and blue subtree has order m
- $F := F(z, u, v)$ suitable generating function
- DEQ $\quad F_z = \frac{v}{(1-z(1+u))^{\frac{1}{1+u}}} e^F$
- Explicit solution $F(z, u, v) = \frac{1}{u} \log \left(\frac{1}{1-v+v(1-z(1+u))^{\frac{u}{1+u}}} \right)$

Parameter I: source degree for Bernoulli model

degree of source



edge-coloured recursive tree

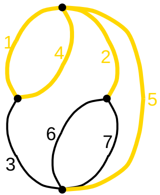


Generating functions approach:

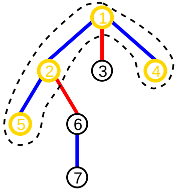
- count ♯ of trees of order n with k blue edges and blue subtree has order m
- $F := F(z, u, v)$ suitable generating function
- DEQ $\quad F_z = \frac{v}{(1-z(1+u))^{\frac{1}{1+u}}} e^F$
- Explicit solution $F(z, u, v) = \frac{1}{u} \log \left(\frac{1}{1-v+v(1-z(1+u))^{\frac{u}{1+u}}} \right)$

Parameter I: source degree for Bernoulli model

degree of source



order of blue subtree

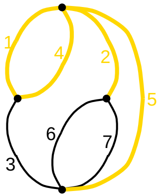


Generating functions approach:

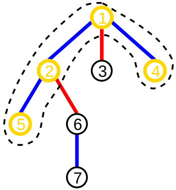
- count $\sharp$ of trees of order n with k blue edges and blue subtree has order m
- $F := F(z, u, v)$ suitable generating function
- DEQ $\quad F_z = \frac{v}{(1-z(1+u))^{\frac{1}{1+u}}} e^F$
- Explicit solution $F(z, u, v) = \frac{1}{u} \log \left(\frac{1}{1-v+v(1-z(1+u))^{\frac{u}{1+u}}} \right)$

Parameter I: source degree for Bernoulli model

degree of source



order of blue subtree

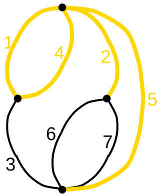


Generating functions approach:

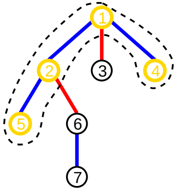
- count $\sharp$ of trees of order n with k blue edges and blue subtree has order m
- $F := F(z,u,v)$ suitable generating function
- DEQ $\quad F_z = \frac{v}{(1-z(1+u))^{\frac{1}{1+u}}} e^F$
- Explicit solution $F(z,u,v) = \frac{1}{u} \log \left(\frac{1}{1-v+v(1-z(1+u))^{\frac{u}{1+u}}} \right)$

Parameter I: source degree for Bernoulli model

degree of source



order of blue subtree



Generating functions approach:

- count $\sharp$ of trees of order n with k blue edges and blue subtree has order m
- $F := F(z,u,v)$ suitable generating function
- DEQ $\quad F_z = \frac{v}{(1-z(1+u))^{\frac{1}{1+u}}} e^F$
- Explicit solution $F(z,u,v) = \frac{1}{u} \log \left(\frac{1}{1-v+v(1-z(1+u))^{\frac{u}{1+u}}} \right)$

Parameter I: source degree for Bernoulli model

D_n : source degree of size-n SP-net

Exact probability distribution [Mahmoud, 2013]:

$$\mathbb{P}\{D_n = m\} = \sum_{j=0}^{m-1} \binom{m-1}{j} (-1)^{n+j-1} \binom{p(j+1)-1}{n-1}$$

Theorem (Kuba & Pan, 2016)

Limiting distribution behaviour: $\frac{D_n}{n^p} \xrightarrow{(d)} D$,

with $D = D(p)$ Mittag-Leffler distribution with parameter p

Sequence of moments $\mathbb{E}(D^r) = \frac{r!}{\Gamma(rp+1)}$, *for* $r \geq 0$

Density function $f(x) = \frac{1}{2\pi i} \int_{\mathcal{H}} \frac{e^{-t-x(-t)^p}}{(-t)^{1-p}} dt$, *for* $x > 0$

Particular instance $p = q = 1/2$: D half-normal distribution

Parameter I: source degree for Bernoulli model

D_n : source degree of size-n SP-net

Exact probability distribution [Mahmoud, 2013]:

$$\mathbb{P}\{D_n = m\} = \sum_{j=0}^{m-1} \binom{m-1}{j} (-1)^{n+j-1} \binom{p(j+1)-1}{n-1}$$

Theorem (Kuba & Pan, 2016)

Limiting distribution behaviour: $\frac{D_n}{n^p} \xrightarrow{(d)} D,$

with $D = D(p)$ Mittag-Leffler distribution with parameter p

Sequence of moments $\mathbb{E}(D^r) = \frac{r!}{\Gamma(rp+1)},$ *for* $r \geq 0$

Density function $f(x) = \frac{1}{2\pi i} \int_{\mathcal{H}} \frac{e^{-t-x(-t)^p}}{(-t)^{1-p}} dt,$ *for* $x > 0$

Particular instance $p = q = 1/2$: D half-normal distribution

Par. II: length of random path for Bernoulli mod.

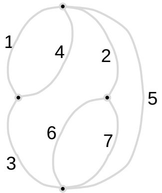
L_n: length of random source-to-sink path in random size-n SP-net

- start at source; at each node x choose one of $d^+(x)$ outgoing edges uniformly at random; until reaching sink

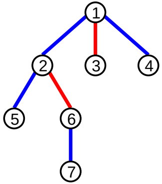
Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path

SP-net



edge-coloured recursive tree



- length of leftmost path corresponds to order of red subtree
- switching colours ⇒ $L_n(p) \stackrel{(d)}{=} D_n(1-p)$

Par. II: length of random path for Bernoulli mod.

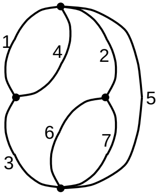
L_n: length of random source-to-sink path in random size-n SP-net

- start at source; at each node x choose one of $d^+(x)$ outgoing edges uniformly at random; until reaching sink

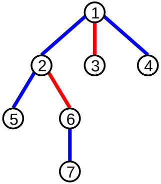
Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path

SP-net



edge-coloured recursive tree



- length of leftmost path corresponds to order of red subtree
- switching colours $\Rightarrow$ $L_n(p) \stackrel{(d)}{=} D_n(1-p)$

Par. II: length of random path for Bernoulli mod.

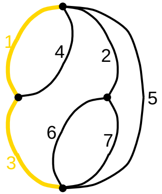
L_n: length of random source-to-sink path in random size-n SP-net

- start at source; at each node x choose one of $d^+(x)$ outgoing edges uniformly at random; until reaching sink

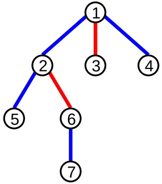
Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path

length of leftmost path



edge-coloured recursive tree



- length of leftmost path corresponds to order of red subtree
- switching colours $\Rightarrow$ $L_n(p) \stackrel{(d)}{=} D_n(1-p)$

Par. II: length of random path for Bernoulli mod.

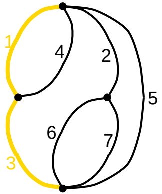
L_n: length of random source-to-sink path in random size-n SP-net

- start at source; at each node x choose one of $d^+(x)$ outgoing edges uniformly at random; until reaching sink

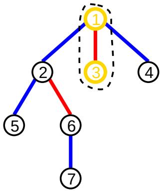
Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path

length of leftmost path



order of red subtree



- length of leftmost path corresponds to order of red subtree
- switching colours $\Rightarrow$ $L_n(p) \stackrel{(d)}{=} D_n(1-p)$

Par. II: length of random path for Bernoulli mod.

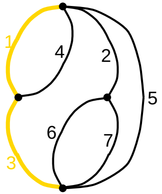
L_n: length of random source-to-sink path in random size-n SP-net

- start at source; at each node x choose one of $d^+(x)$ outgoing edges uniformly at random; until reaching sink

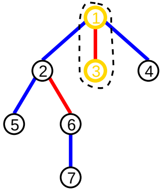
Observations:

- $L_n \overset{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path

length of leftmost path



order of red subtree



- length of leftmost path corresponds to order of red subtree
- switching colours $\Rightarrow$ $L_n(p) \overset{(d)}{=} D_n(1-p)$

Par. II: length of random path for Bernoulli mod.

L_n : length of random source-to-sink path in random size-n SP-net

Theorem (Kuba & Pan, 2016)

Exact probability distribution:

$$\mathbb{P}\{L_n = m\} = \sum_{j=0}^{m-1} \binom{m-1}{j} (-1)^{n+j-1} \binom{j - p(j+1)}{n-1}$$

Limiting distribution behaviour: $\frac{L_n}{n^{1-p}} \xrightarrow{(d)} L$,

with $L = L(p)$ Mittag-Leffler distribution with parameter $1 - p$

Sequence of moments $\mathbb{E}(L^r) = \dfrac{r!}{\Gamma(r(1-p)+1)}$, *for $r \geq 0$*

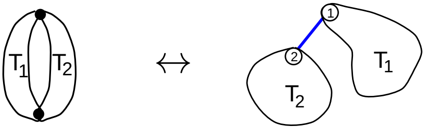
Density function $f(x) = \dfrac{1}{2\pi i} \displaystyle\int_{\mathcal{H}} \frac{e^{-t-x(-t)^{1-p}}}{(-t)^p} dt$, *for $x > 0$*

Par. III: number of paths for Bernoulli model

Source-to-sink paths according to first edge-duplication:

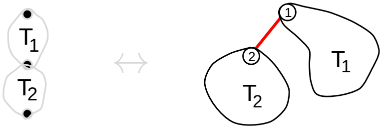
parallel duplication: $paths(T) = paths(T_1) + paths(T_2)$

T :



serial duplication: $paths(T) = paths(T_1) \cdot paths(T_2)$

T :



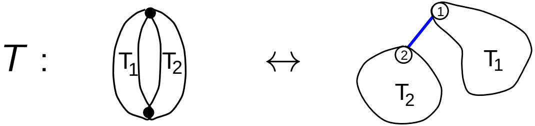
P_n: number of source-to-sink paths in random size-n SP-net

- stochastic recurrence for P_n via tree decomposition
- Bernoulli-DEQ for g.f. of expectation $\mathbb{E}(P_n)$
  $\rightarrow$ exact/asymptotic solution for $\mathbb{E}(P_n)$

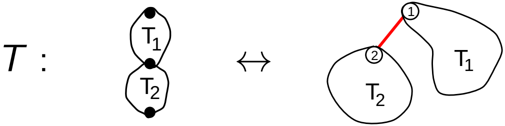
Par. III: number of paths for Bernoulli model

Source-to-sink paths according to first edge-duplication:

parallel duplication: $paths(T) = paths(T_1) + paths(T_2)$



serial duplication: $paths(T) = paths(T_1) \cdot paths(T_2)$



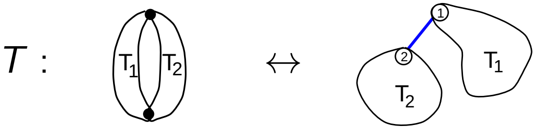
P_n: number of source-to-sink paths in random size-n SP-net

- stochastic recurrence for P_n via tree decomposition
- Bernoulli-DEQ for g.f. of expectation $\mathbb{E}(P_n)$
  $\rightarrow$ exact/asymptotic solution for $\mathbb{E}(P_n)$

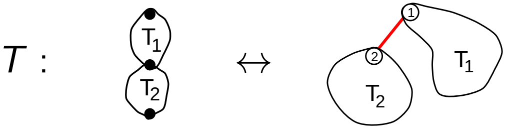
Par. III: number of paths for Bernoulli model

Source-to-sink paths according to first edge-duplication:

parallel duplication: $paths(T) = paths(T_1) + paths(T_2)$



serial duplication: $paths(T) = paths(T_1) \cdot paths(T_2)$



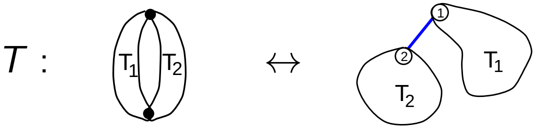
P_n: number of source-to-sink paths in random size-n SP-net

- stochastic recurrence for P_n via tree decomposition
- Bernoulli-DEQ for g.f. of expectation $\mathbb{E}(P_n)$
  $\rightarrow$ exact/asymptotic solution for $\mathbb{E}(P_n)$

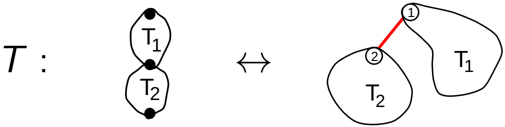
Par. III: number of paths for Bernoulli model

Source-to-sink paths according to first edge-duplication:

parallel duplication: $paths(T) = paths(T_1) + paths(T_2)$



serial duplication: $paths(T) = paths(T_1) \cdot paths(T_2)$



P_n: number of source-to-sink paths in random size-n SP-net

- stochastic recurrence for P_n via tree decomposition
- Bernoulli-DEQ for g.f. of expectation $\mathbb{E}(P_n)$
  $\rightarrow$ exact/asymptotic solution for $\mathbb{E}(P_n)$

Par. III: number of paths for Bernoulli model

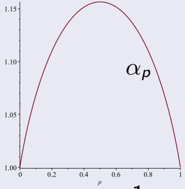
P_n: number of source-to-sink paths in random size-n SP-net

Theorem (Kuba & Pan, 2016)

Asymptotic behaviour of expectation $\mathbb{E}(P_n)$:

$$\mathbb{E}(P_n) = \frac{1}{1-p} \cdot \alpha_p^n + R_p(n),$$

$$\textit{where } \alpha_p = \begin{cases} \frac{1}{1-\left(\frac{p}{1-p}\right)^{\frac{1}{1-2p}}}, & \textit{for } p \neq \frac{1}{2}, \\ \frac{1}{1-e^{-2}}, & \textit{for } p = \frac{1}{2} \end{cases}$$



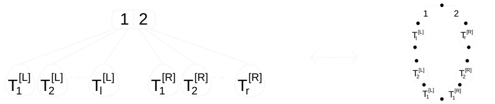
$$\textit{and } R_p(n) = \begin{cases} -\frac{1-2p}{p\Gamma(2p)} n^{2p-1} + \mathcal{O}(n^{2(2p-1)}), & \textit{for } 0 < p < \frac{1}{2}, \\ -\frac{2}{\log n} + \mathcal{O}(\frac{1}{\log^2 n}), & \textit{for } p = \frac{1}{2}, \\ -\frac{2p-1}{1-p} + \mathcal{O}(n^{1-2p}), & \textit{for } \frac{1}{2} < p < 1 \end{cases}$$

Par. IV: length of random path for Binary mod.

L_n: length of random source-to-sink path in random size-n SP-net

Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path
- subblock structure $\rightarrow$ recursive description

$T:$ 

$\rightarrow \quad length(T) = 1 + \underbrace{length(T_1^{[L]}) + \cdots + length(T_\ell^{[L]})}_{\text{sum of lengths in left forest}}$

Generating functions approach:

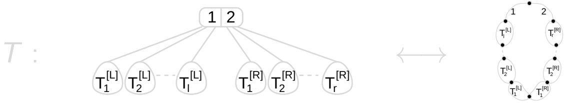
- apply combinatorial decomposition
- g.f. $F(z, v) \rightarrow$ non-linear DEQ: $F''(z, v) = \frac{v}{1-z} e^{F(z,v)}$
- apply method of moments

Par. IV: length of random path for Binary mod.

L_n: length of random source-to-sink path in random size-n SP-net

Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path
- subblock structure → recursive description



$$\rightarrow \quad length(T) = 1 + \underbrace{length(T_1^{[L]}) + \cdots + length(T_\ell^{[L]})}_{\text{sum of lengths in left forest}}$$

Generating functions approach:

- apply combinatorial decomposition
- g.f. $F(z,v) \to$ non-linear DEQ: $F''(z,v) = \frac{v}{1-z} e^{F(z,v)}$
- apply method of moments

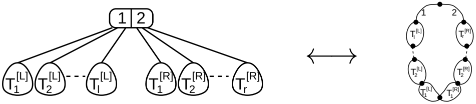
Par. IV: length of random path for Binary mod.

L_n: length of random source-to-sink path in random size-n SP-net

Observations:

- $L_n \stackrel{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path
- subblock structure $\rightarrow$ recursive description

T :



$$\rightarrow \quad length(T) = 1 + \underbrace{length(T_1^{[L]}) + \cdots + length(T_\ell^{[L]})}_{\text{sum of lengths in left forest}}$$

Generating functions approach:

- apply combinatorial decomposition
- g.f. $F(z,v) \rightarrow$ non-linear DEQ: $F''(z,v) = \frac{v}{1-z} e^{F(z,v)}$
- apply method of moments

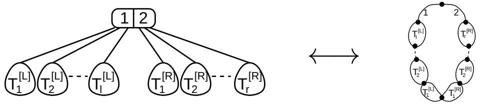
Par. IV: length of random path for Binary mod.

L_n: length of random source-to-sink path in random size-n SP-net

Observations:

- $L_n \overset{(d)}{=} L_n^{[left]}$: length of leftmost source-to-sink path
- subblock structure $\rightarrow$ recursive description

T :



$$\rightarrow \qquad length(T) = 1 + \underbrace{length(T_1^{[L]}) + \cdots + length(T_\ell^{[L]})}_{\text{sum of lengths in left forest}}$$

Generating functions approach:

- apply combinatorial decomposition
- g.f. $F(z,v) \rightarrow$ non-linear DEQ: $F''(z,v) = \frac{v}{1-z} e^{F(z,v)}$
- apply method of moments

Par. IV: length of random path for Binary mod.

L_n: length of random source-to-sink path in random size-n SP-net

Expectation [Mahmoud, 2013]:

$$\mathbb{E}(L_n) = n\left(\frac{3+\sqrt{5}}{2\sqrt{5}}\binom{n+\frac{\sqrt{5}}{2}-\frac{3}{2}}{n} - \frac{3-\sqrt{5}}{2\sqrt{5}}\binom{n-\frac{\sqrt{5}}{2}-\frac{3}{2}}{n}\right) \sim \frac{1+\sqrt{5}}{2\sqrt{5}}\frac{n^{\frac{\sqrt{5}-1}{2}}}{\Gamma(\frac{\sqrt{5}-1}{2})}$$

Theorem (Kuba & Pan, 2016)

Limiting distribution behaviour: $\frac{L_n}{n^{\phi}} \xrightarrow{(d)} L, \quad \phi = \frac{\sqrt{5}-1}{2},$

with L characterized by sequence of r-th integer moments:

$$\mathbb{E}(L^r) = \frac{r!\cdot c_r}{\Gamma(r\phi+1)}, \quad r \geq 0,$$

where sequence $(c_r)_r$ satisfies recurrence $c_0 = 1$, $c_1 = \frac{3+\phi}{5}$,

$$c_r = \frac{1}{\phi(r-1)((r+1)\phi+1)}\sum_{k=1}^{r-1}(k\phi+1)c_k c_{r-k}, \quad \textit{for } r \geq 2$$

Par. IV: length of random path for Binary mod.

L_n: length of random source-to-sink path in random size-n SP-net

Expectation [Mahmoud, 2013]:

$$\mathbb{E}(L_n) = n\left(\frac{3+\sqrt{5}}{2\sqrt{5}}\binom{n+\frac{\sqrt{5}}{2}-\frac{3}{2}}{n} - \frac{3-\sqrt{5}}{2\sqrt{5}}\binom{n-\frac{\sqrt{5}}{2}-\frac{3}{2}}{n}\right) \sim \frac{1+\sqrt{5}}{2\sqrt{5}}\frac{n^{\frac{\sqrt{5}-1}{2}}}{\Gamma(\frac{\sqrt{5}-1}{2})}$$

Theorem (Kuba & Pan, 2016)

Limiting distribution behaviour: $\frac{L_n}{n^\phi} \xrightarrow{(d)} L, \quad \phi = \frac{\sqrt{5}-1}{2}$,

with L characterized by sequence of r-th integer moments:

$$\mathbb{E}(L^r) = \frac{r!\cdot c_r}{\Gamma(r\phi+1)}, \quad r \geq 0,$$

where sequence $(c_r)_r$ satisfies recurrence $c_0 = 1$, $c_1 = \frac{3+\phi}{5}$,

$$c_r = \frac{1}{\phi(r-1)((r+1)\phi+1)}\sum_{k=1}^{r-1}(k\phi+1)c_k c_{r-k}, \quad \text{for } r \geq 2$$

Further Parameters for Binary model

Theorem (Kuba & Pan, 2016)

D_n: *degree of sink in random size-n SP-net*

Limiting distribution behaviour: $\frac{D_n}{n^{\sqrt{2}-1}} \xrightarrow{(d)} D$,

$$\mathbb{E}(D^r) = \frac{r!(r(\sqrt{2}-1)+1)c_r}{\Gamma(r(\sqrt{2}-1)+1)}, \quad r \geq 0,$$

where sequence $(c_r)_{r\geq 0}$ satisfies "convolution-type recurrence"

Theorem (Kuba & Pan, 2016)

P_n: *number of source-to-sink paths in random size-n SP-net*

Expectation: $\mathbb{E}(P_n) = \frac{2}{\rho^n} \cdot \left(1 - \frac{\rho^2}{(\rho-1)^2(n-1)(n-2)} + \mathcal{O}\left(\frac{\log n}{n^4}\right)\right)$,

with $\rho \approx 0.89\ldots$

Further Parameters for Binary model

Theorem (Kuba & Pan, 2016)

D_n: *degree of sink in random size-n SP-net*

Limiting distribution behaviour: $\frac{D_n}{n^{\sqrt{2}-1}} \xrightarrow{(d)} D,$

$$\mathbb{E}(D^r) = \frac{r!(r(\sqrt{2}-1)+1)c_r}{\Gamma(r(\sqrt{2}-1)+1)}, \quad r \geq 0,$$

where sequence $(c_r)_{r\geq 0}$ satisfies "convolution-type recurrence"

Theorem (Kuba & Pan, 2016)

P_n: *number of source-to-sink paths in random size-n SP-net*

Expectation: $\mathbb{E}(P_n) = \frac{2}{\rho^n} \cdot \left(1 - \frac{\rho^2}{(\rho-1)^2(n-1)(n-2)} + \mathcal{O}\left(\frac{\log n}{n^4}\right)\right),$

with $\rho \approx 0.89\ldots$

Relations to other work

- **Increasing diamonds:**
  - SP-net model introduced by [Bodini, Dien, Fontaine, Genitrini and Hwang, 2016]
  - different structure, study yields second-order non-linear DEQs
- **Mittag-Leffler limiting distributions** in comb. contexts:
  - triangular balanced urn models [Janson, 2010]
  - extra clustering model for animal grouping [Drmota, Fuchs and Lee, 2015]
- **Recurrences of "convolution type":**
  similar to ones studied via Psi-series method in [Chern, Fernández-Camacho, Hwang and Martínez, 2014]
- **Top-down decomposition:**
  maybe alternative characterizations via contraction method, e.g., [Neininger and Rschendorf, 2004]

Relations to other work

- **Increasing diamonds:**
  - SP-net model introduced by [Bodini, Dien, Fontaine, Genitrini and Hwang, 2016]
  - different structure, study yields second-order non-linear DEQs
- **Mittag-Leffler limiting distributions** in comb. contexts:
  - triangular balanced urn models [Janson, 2010]
  - extra clustering model for animal grouping [Drmota, Fuchs and Lee, 2015]
- **Recurrences of "convolution type":**
  similar to ones studied via Psi-series method in [Chern, Fernández-Camacho, Hwang and Martínez, 2014]
- **Top-down decomposition:**
  maybe alternative characterizations via contraction method, e.g., [Neininger and Rschendorf, 2004]

Relations to other work

- **Increasing diamonds:**
  - SP-net model introduced by [Bodini, Dien, Fontaine, Genitrini and Hwang, 2016]
  - different structure, study yields second-order non-linear DEQs
- **Mittag-Leffler limiting distributions** in comb. contexts:
  - triangular balanced urn models [Janson, 2010]
  - extra clustering model for animal grouping [Drmota, Fuchs and Lee, 2015]
- **Recurrences of “convolution type”:**

  similar to ones studied via Psi-series method in [Chern, Fernández-Camacho, Hwang and Martínez, 2014]
- **Top-down decomposition:**

  maybe alternative characterizations via contraction method, e.g., [Neininger and Rschendorf, 2004]

Relations to other work

- **Increasing diamonds:**
  - SP-net model introduced by [Bodini, Dien, Fontaine, Genitrini and Hwang, 2016]
  - different structure, study yields second-order non-linear DEQs
- **Mittag-Leffler limiting distributions** in comb. contexts:
  - triangular balanced urn models [Janson, 2010]
  - extra clustering model for animal grouping [Drmota, Fuchs and Lee, 2015]
- **Recurrences of "convolution type":**

  similar to ones studied via Psi-series method in [Chern, Fernández-Camacho, Hwang and Martínez, 2014]
- **Top-down decomposition:**

  maybe alternative characterizations via contraction method, e.g., [Neininger and Rschendorf, 2004]

Outlook

Study further quantities via combinatorial decomposition:

- number of ancestors of nodes/edges
- number of descendants of nodes/edges
- node degrees
- number of spanning trees
- number of independent sets

Further stochastic SP-net models with comb. description:

- b-ary saturation model for SP-nets $\leftrightarrow$ bucket recursive trees with bucket-size $b \geq 2$
- "preferential edge-duplication rule" for Bernoulli model $\leftrightarrow$ edge-coloured plane increasing trees

Outlook

Study further quantities via combinatorial decomposition:

- number of ancestors of nodes/edges
- number of descendants of nodes/edges
- node degrees
- number of spanning trees
- number of independent sets

Further stochastic SP-net models with comb. description:

- b-ary saturation model for SP-nets $\leftrightarrow$ bucket recursive trees with bucket-size $b \geq 2$
- "preferential edge-duplication rule" for Bernoulli model $\leftrightarrow$ edge-coloured plane increasing trees