

## § 17 Bounding the distribution function of the makespan

### Approximate methods and bounds

#### Simulation

- ◆ requires (complete) information about distributions
- ◆ difficult to model stochastic dependencies

#### Bounding the distribution function

- ◆ possible with incomplete information
- ◆ permits stochastic dependencies
- ◆ combinatorial algorithms

↑  
here, makes heavy use of arc diagrams

### Chain minor

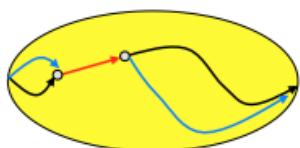
Let  $N, N'$  be project networks. (as arc diagrams)

$N$  is a **chain minor** of  $N'$  if

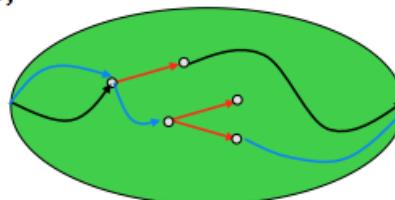
- every job of  $N$  is **represented** by (one or several) copies in  $N'$
- every chain of  $N$  is **contained** in a chain of  $N'$

by taking an appropriate copy

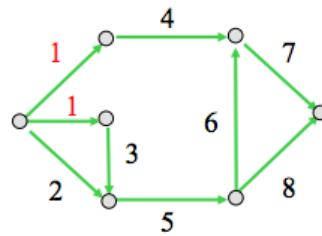
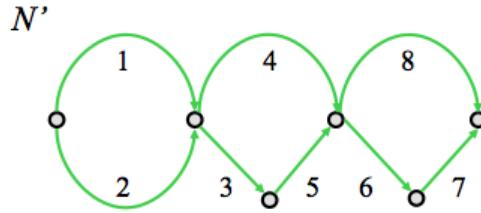
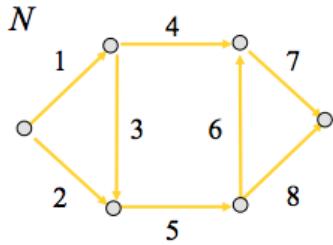
$N$



$N'$

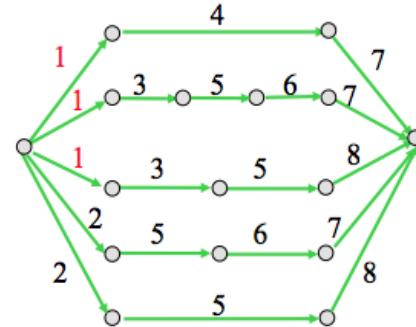
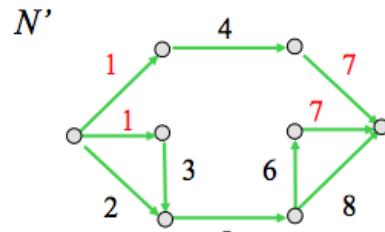
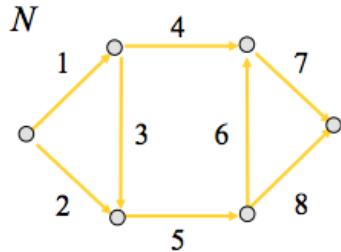


## Chain minors: an example



no copies of job  
this gives an extension  
of G

job 1 represented  
by 2-copies,  
equality of chains



extreme case:  
parallel composition  
of all chains

## Bounds based on chain minors

M. & Müller '99

- ▶ Let  $N$  be a chain minor of  $N'$
- Give copies of a job from  $N$  in  $N'$  the same processing time distribution as their original
- Take all processing time distributions in  $N'$  as stochastically independent
- ▶ Then the makespan of  $N$  is stochastically smaller than the makespan of  $N'$

$$Q_{C_{\max}(N)} \leq_{st} Q_{C_{\max}(N')}$$

*needs definition*

17.1 Theorem

Let  $X_1, X_2$  real-valued random variables with distributions  $P_1$  and  $P_2$   
and distribution functions  $F_1$  and  $F_2$

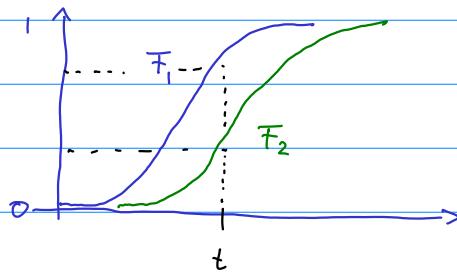
Then  $X_1$  is stochastically smaller than  $X_2$

$$P_1 \dots$$

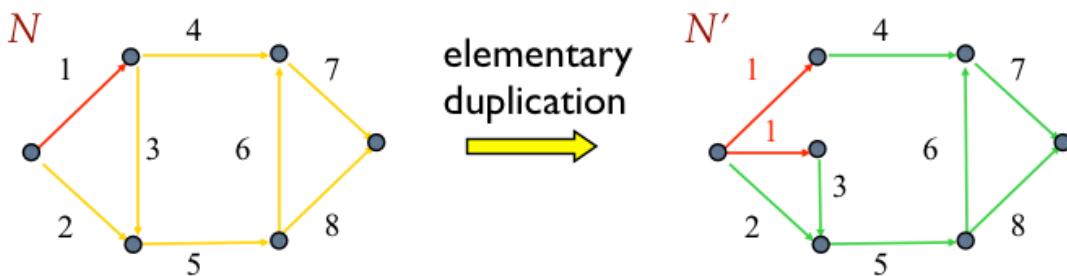
$$P_2$$

If  $F_1 \geq F_2$  pointwise, i.e.  $\text{Prob}\{X_1 \leq t\} \geq \text{Prob}\{X_2 \leq t\} \quad \forall t$   
 $\Leftrightarrow \int f dP_1 \leq \int f dP_2 \quad \forall f: \mathbb{R}^1 \rightarrow \mathbb{R}^1 \text{ non-decreasing}$

Intuition:  $X_1$  is smaller than  $X_2$  with higher probability



## Intuition for chain minor bounding principle



copies have same processing time as original  $\Rightarrow$  same makespan

independent variation increases makespan stochastically, i.e.

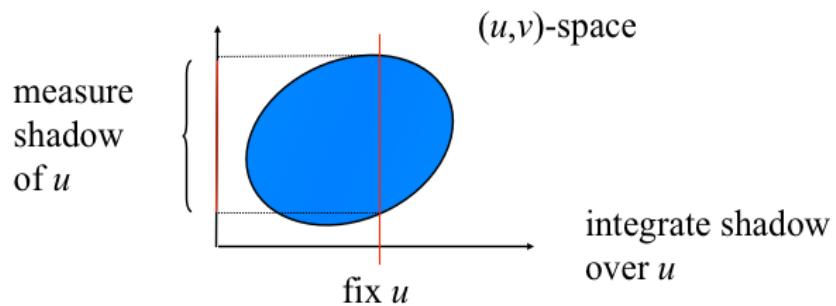
$$\Pr(C_{\max}(N) \leq t) \geq \Pr(C_{\max}(N') \leq t) \text{ for all } t$$

Proof: It suffices to consider only one duplication

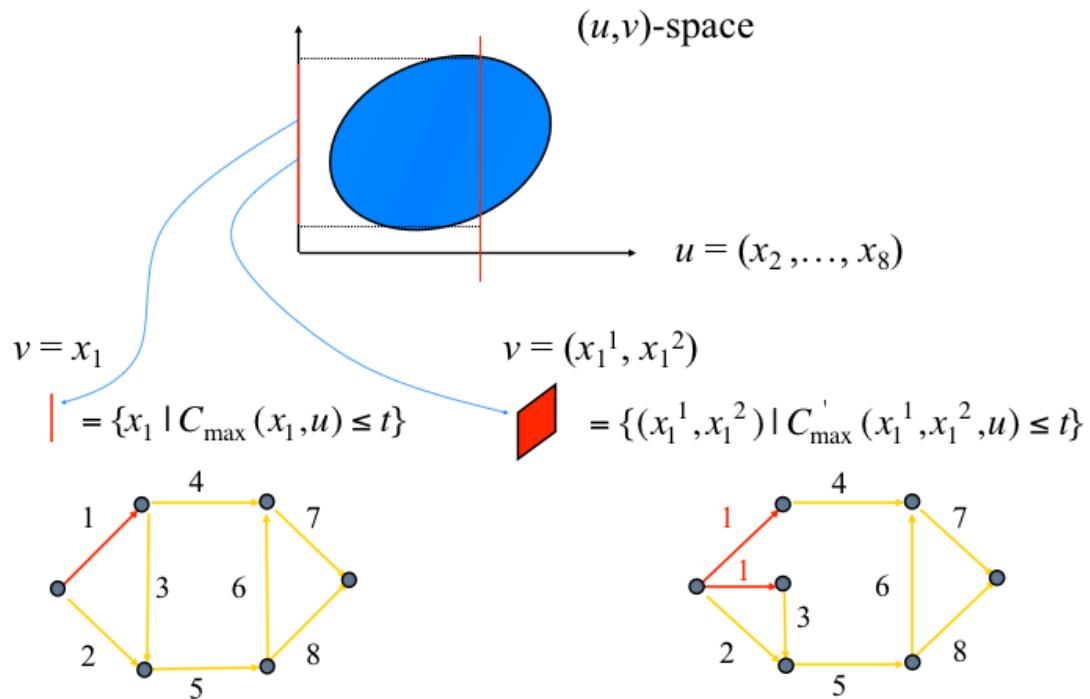
The argument can be repeated if there are more duplications.

The proof is done on an example network, but it is completely generic

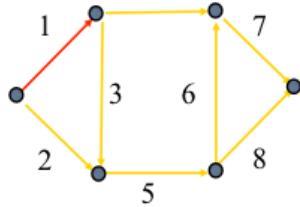
## Sketch of proof



$$\Pr\{C_{\max}(N) \leq t\} \geq \Pr\{C_{\max}(N') \leq t\}$$



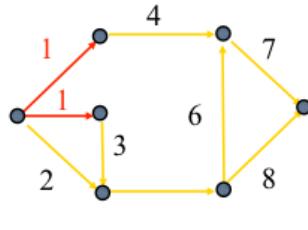
$$Q_1(|)$$



$$| = [0, a]$$

$\geq$

$$Q_1 \otimes Q_1(\blacksquare)$$



$$\blacksquare = [0, b] \times [0, c]$$

Assume  $b \geq c$ . Then  $a \geq c$ .

←

see argument below

$$Q_1(|) = Q_1([0, a])$$

$$\geq Q_1([0, c])$$

$$\geq Q_1([0, b]) Q_1([0, c])$$

$$= Q_1 \otimes Q_1([0, b] \times [0, c]) = Q_1 \otimes Q_1(\blacksquare)$$

← since  $Q_1[0, b] \leq 1$

↑

use independence

Arguments for the pictures:

Let  $(x_1, \dots, x_8)$  be a vector of processing times for  $N$

Let  $(x'_1, x^2_1, x_2, \dots, x_8)$  the vector after duplication in  $N'$

Set  $u := (x_2, \dots, x_8)$ , the "joint processing times"

Let  $A := \{x_i \mid C_{\max}(x_i, u) \leq t\}$  in  $N$

$B := \{(x'_1, x^2_1) \mid C_{\max}(x'_1, x^2_1, u) \leq t\}$  in  $N'$

$\Rightarrow A$  is a 1-dimensional interval, say  $A = [0, a]$

$B$  is a 2-dimensional interval, say  $B = [0, b] \times [0, c]$

since the 2 copies of job 1 can take values independently

Let w.o.l.g.  $b > c$ .

Then  $a \geq c$

If not, then  $a < c$

$\Rightarrow (c, u)$  gives  $C_{\max}(c, u) > t$  in  $N$

but  $C_{\max}(c, c, u) \leq t$  in  $N'$

a contradiction to  $C_{\max}^N(c, u) = C_{\max}^{N'}(c, c, u)$

So for every fixed  $u = (x_2, \dots, x_8)$  we have

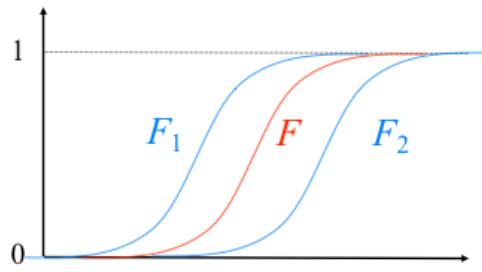
$$Q_1(A) \geq Q_1 \otimes Q_1(B)$$

Integration over  $u$  gives  $\Pr(C_{\max}^N \leq t) \geq \Pr(C_{\max}^{N'} \leq t)$   $\square$

## Sandwiching a network with minors

Given  $N$ , look for **special networks**  $N_1, N_2$ , such that

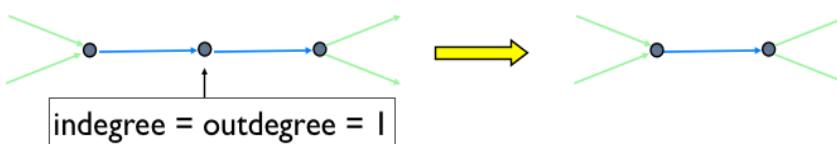
$$N_1 \subseteq_{\text{minor}} N \subseteq_{\text{minor}} N_2$$



**special** = easier to calculate the makespan distribution

## Series-parallel networks

**series reduction**



**parallel reduction**



A network  $N$  is **series-parallel** if it can be reduced by a finite sequence of series and parallel reductions to one job.

← one of many  
equivalent definitions

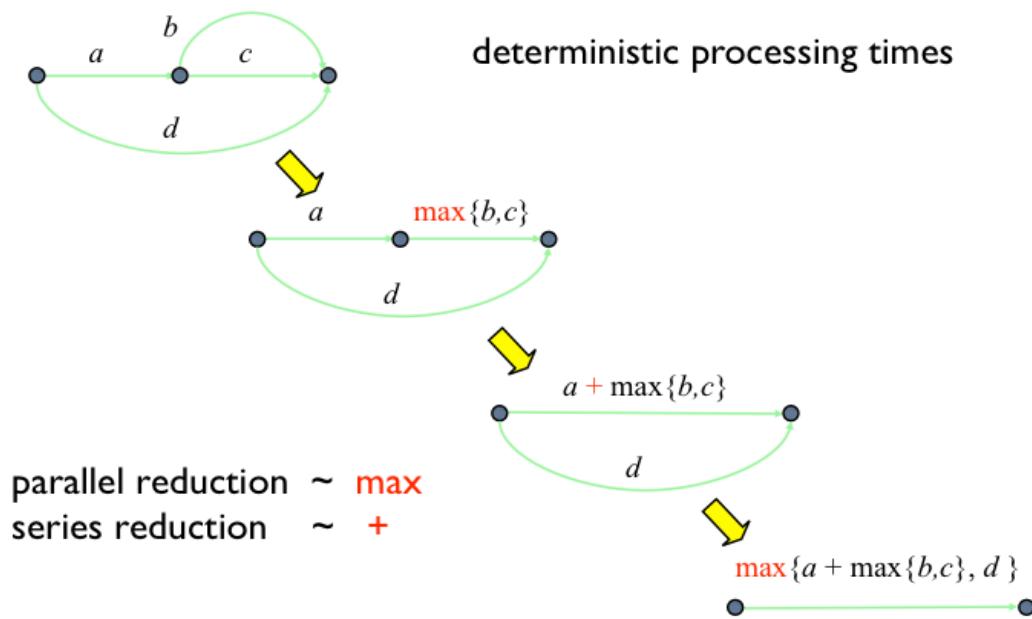
# Properties of series-parallel networks

Valdes, Tarjan & Lawler '82

- ◆ Series-parallel networks can be recognized in linear time
- ◆ A reduction sequence can be constructed in linear time

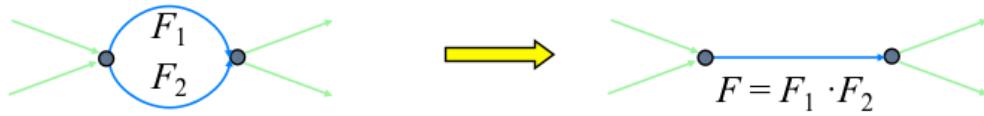
- ◆ Makespan and makespan distribution can be calculated along a reduction sequence

## Makespan computation along a reduction sequence

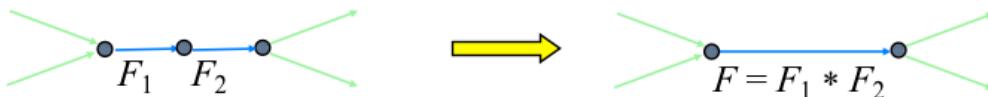


## Makespan distribution of a series-parallel network

parallel reduction ~ **product** of distribution functions



series reduction ~ **convolution** of distribution functions



$$F_1 * F_2(t) = \int_0^t F_1(t-x) f_2(x) dx \quad f_2 = \text{density of } F_2$$

Evaluating the distribution function of series-parallel networks is indeed easier, but still NP-hard in the weak sense

## Complexity of evaluating series-parallel networks

M. & Müller '99

- (1) ▶ DF is NP-hard (in the weak sense) for series-parallel networks with 2-point processing time distributions

But:

- (2) ▶ The reduction sequence algorithm computes the makespan distribution function in a number of steps polynomial in  
job processing times )  
 $M = \max \#(\text{values of the makespan})$   
along the sequence for discrete distributions

17.2 THEOREM

Proof: (1) Proof by reduction from PARTITION

let  $a_1, \dots, a_n, b \in \mathbb{N}$  be an instance I of PARTITION

↑

$2b = \sum a_i$ , Question: Is there a subset that adds up to b

Construct an instance  $I'$  of DF-SP as follows:



Job j has processing times  $X_j = \begin{cases} 0 & \text{with prob } \frac{1}{2} \\ a_j & \text{with prob } \frac{1}{2} \end{cases}$

Assume there is a polynomial algorithm A that computes  $\bar{F}_{C_{\max}}(t)$  for arbitrary t

Then: use A to compute  $\bar{F}_{C_{\max}}(t)$  for  $t_1 = b$ ,  $t_2 = b-1$

If I is a yes-instance of PARTITION

$\Leftrightarrow \{C_{\max} = b\}$  has positive probability

$\Leftrightarrow \bar{F}_{C_{\max}}$  has a jump at  $t = b$

$\Leftrightarrow$  the values  $\bar{F}_{C_{\max}}(b)$  and  $\bar{F}_{C_{\max}}(b-1)$  differ

So we can decide in polynomial time if I is a yes-instance of PARTITION

↑

Note: this is a Turing reduction

(2) Let  $N := \max \# \text{values of jobs along a reduction sequence}$

[Then we will show a run-time of  $O(N^2 \cdot n)$ ]

Assume that  $\bar{F}_j$  of job j is represented by a list  $t_1 < t_2 < \dots < t_\ell$  of its jumps with values  $F_j(t_1), \dots, F_j(t_\ell)$

$\Rightarrow \bar{F}_j(t)$  can be calculated by a scan through the list in  $O(\ell)$  time

assumption  $\Rightarrow \ell \leq N$  at every stage of the reduction sequence

$\Rightarrow$  computing  $\bar{F}_i \cdot \bar{F}_j$  takes  $O(N)$  time

computing  $\bar{F}_i * \bar{F}_j$  takes  $O(N^2)$  time

length of a reduction sequence =  $n-1 \Rightarrow O(N^2 n)$  in total □

## Specific Bounds

(A)

### The bound of Dodin '85

Input: Stochastic project network  $N$

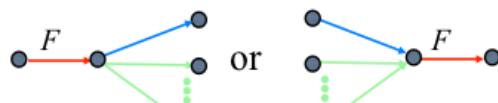
Output: An upper bound on the makespan distribution of  $N$

**while**  $N$  has more than one arc **do**

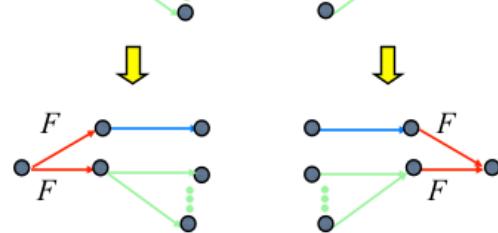
if a series reduction is possible then apply one

**else if** a parallel reduction is possible **then** apply one

**else** find



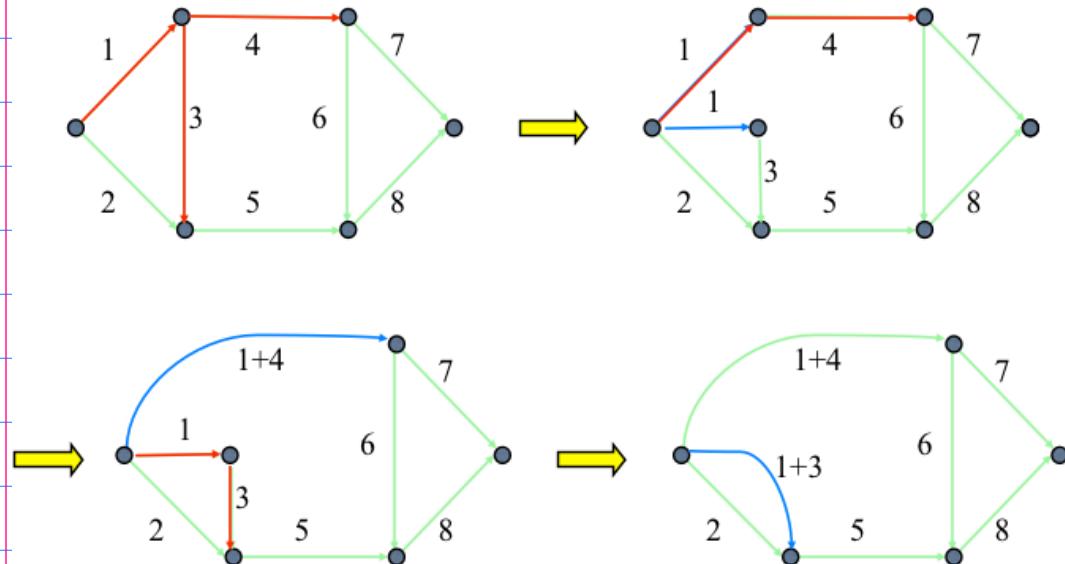
duplicate

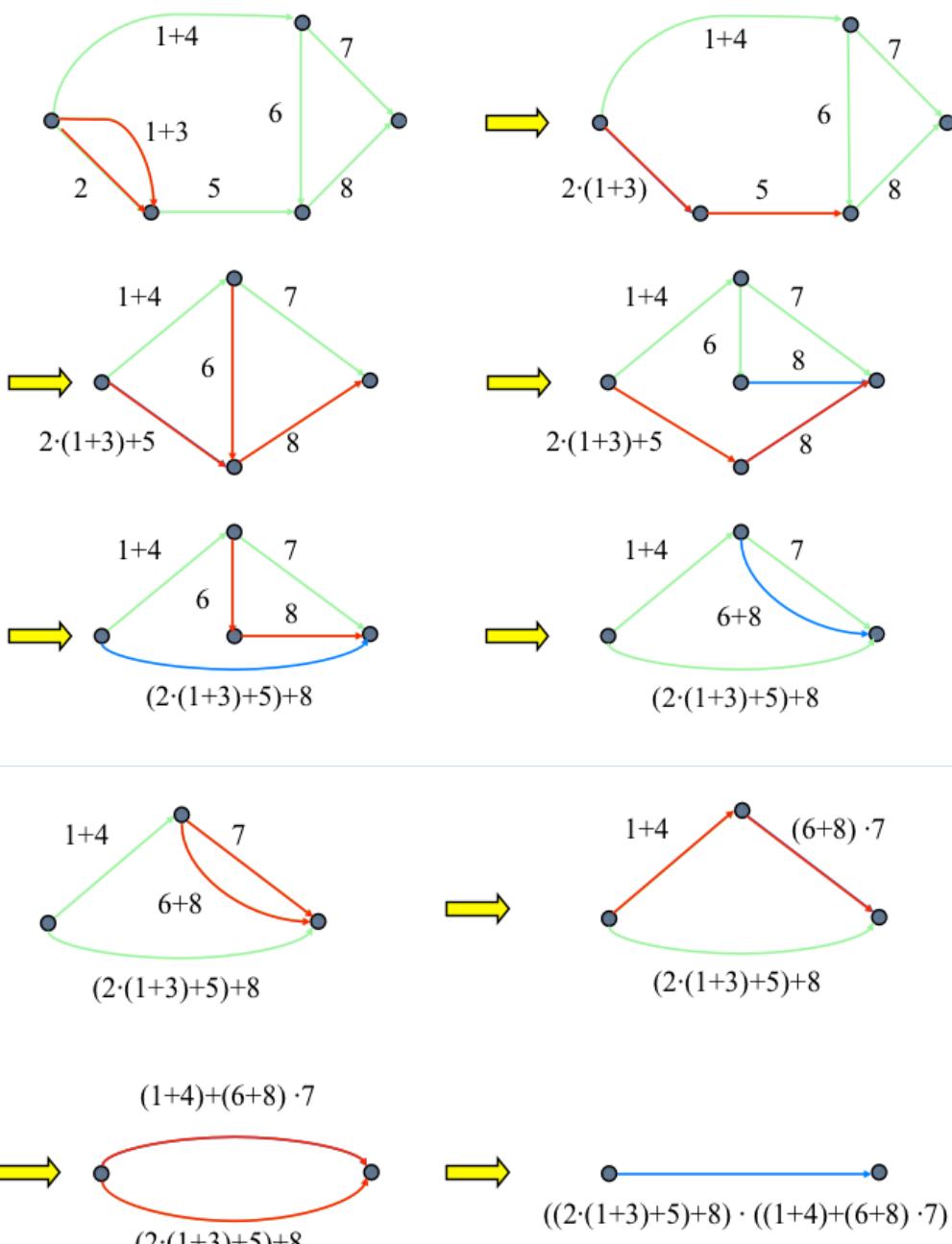


Exercise: show that Dodin's algorithm always terminates with a single job.

The distribution of that job is an upper bound for  $\bar{F}_{\text{max}}^N$

### An example of Dodin's algorithm





resulting distribution function:

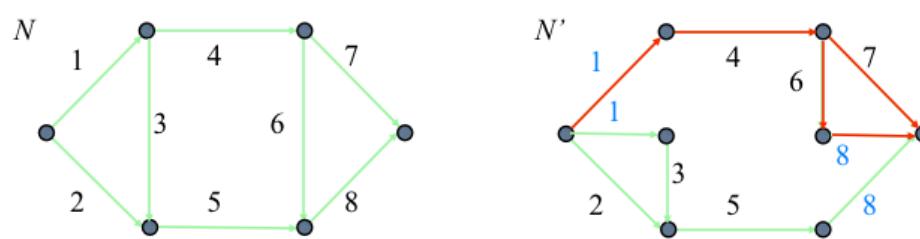
$$((F_2 \cdot (F_1 * F_3) * F_5) * F_8) \cdot ((F_1 * F_4) * (F_6 * F_8) * F_7)$$

## Structure of Dodin's bound

Dodin's bound for network  $N$  is

- ▶ exact if  $N$  is series-parallel
- ▶ the distribution function of a series-parallel network  $N'$  that contains  $N$  as a chain-minor

$$((F_2 \cdot (F_1 * F_3) * F_5) * F_8) \cdot ((F_1 * F_4) * (F_6 * F_8) * F_7)$$



Proof: clear from the bounding principle for chain minors  $\square$

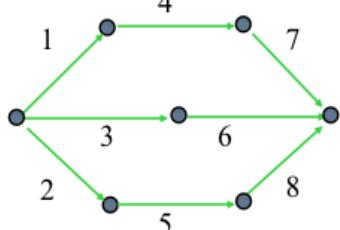
(B)

## The bounds of Spelde '76

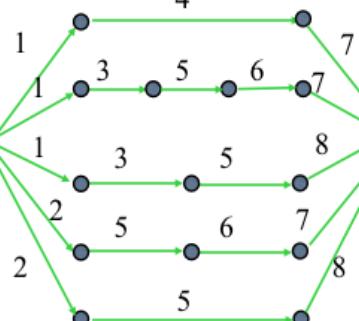
- ▶ uses special series-parallel networks (parallel chains)
- ▶ yields upper and lower bounds

$\leftarrow$  clear from minor property

disjoint chains  
⇒ lower bound



all chains  
⇒ upper bound



# Distribution-free evaluation of Spelde's bounds

Large networks  $\Rightarrow$

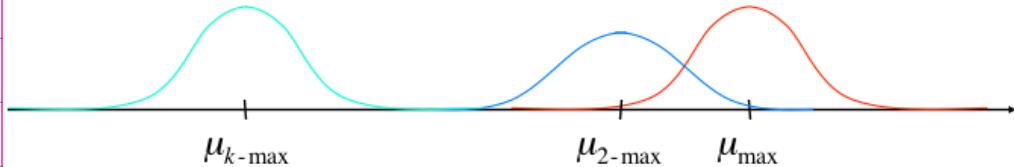
- ▶ chain length  $\approx$  normally distributed
  - $\Rightarrow \mu = \sum \mu_j$  and  $\sigma^2 = \sum \sigma_j^2$  along a chain
- ▶  $\Rightarrow$  per job only  $\mu_j$  and  $\sigma_j^2$  required

$$\begin{aligned} \mu_j &= \text{mean} \\ \sigma_j^2 &= \text{variance} \end{aligned} \quad \left. \begin{array}{l} \text{of job } j \end{array} \right\}$$

Problem: #chains may be exponential

Remedie: Consider only **relevant** chains

## Relevant chains



$$\Pr(Y_k \geq Y_1) \leq \varepsilon \quad Y_i = \text{length of } i\text{-longest chain} \quad \text{w.r.t. mean processing times } \mu_j$$

Apply  $k$ -longest path algorithms to determine  $k$

$\leftarrow$  see (\*), (\*\*), below

Return the product of the  $k-1$  normal distribution functions  $F_{Y_i}$

Excellent and fast bounds in practice

Special case: **PERT**, considers only  $Y_1$

(\*) Compute 1st, 2nd, ...,  $k$ -th longest chain w.r.t.  $\mu_j$   
until  $\Pr\{Y_k \geq Y_1\} \leq \varepsilon$  for a specified  $\varepsilon$

$\uparrow \quad \uparrow$   
length of chains as normal distribution

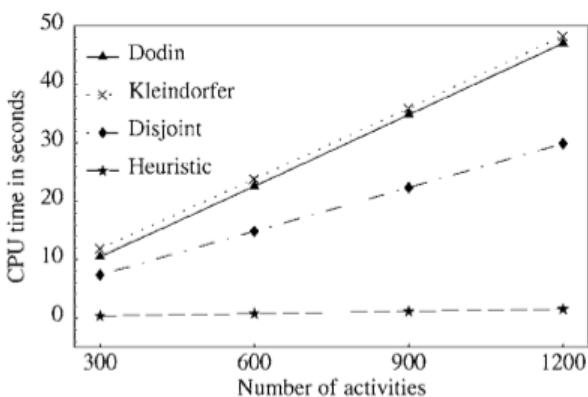
Then  $\bar{T} := \bar{T}_1 \cdot \bar{T}_2 \cdot \dots \cdot \bar{T}_{k-1}$  is an upper bound for  $\bar{T}_{C_{\max}^N}$

(\*\*) To compute the  $k$ -longest paths one may use a  $k$ -shortest path algorithm, since  $N$  is acyclic

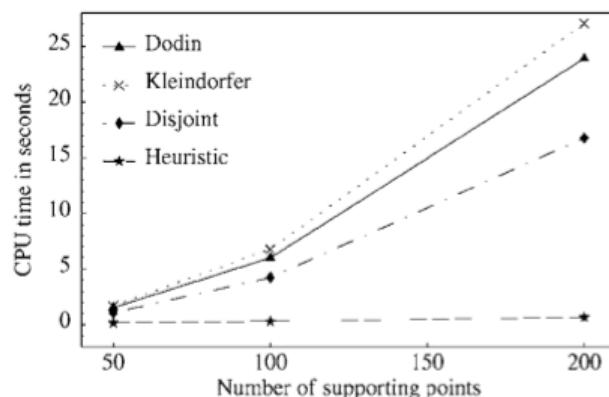
Such algorithms exist and run in  $O(k \cdot n(m + n \cdot \log n))$

see: [www.ics.uci.edu/~eppstein/bibs/kpath.bib](http://www.ics.uci.edu/~eppstein/bibs/kpath.bib)

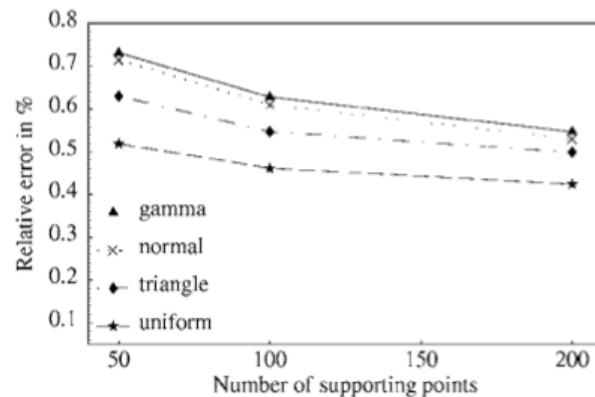
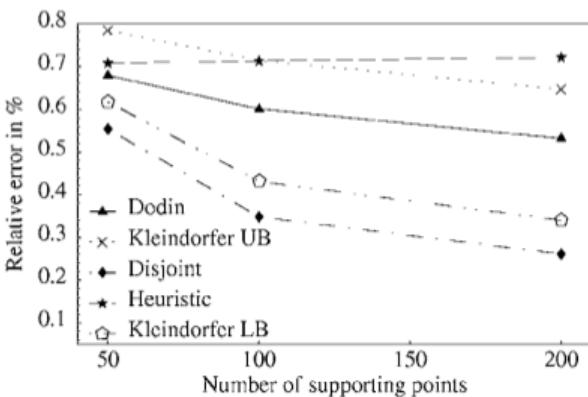
Computational experience with Dodin, Spelde and other bounds ( $\sim 2000$ )



(a)



(b)



Heuristic  $\hat{=}$  Spelde distribution free

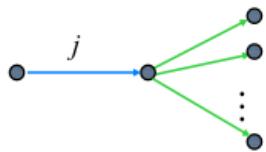
number of supporting points = representation of discrete distribution function

relative error  $\hat{=}$  in comparison with a very precise simulation

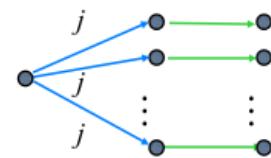
The "distance" of a partial order to a series-parallel order determines the quality of the bounds in Dodin's algorithm.

Appropriate distance measures have been explored in combinatorics

## Measuring the distance to series-parallel networks

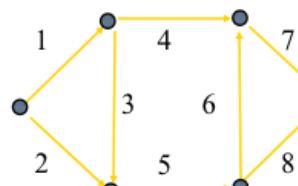


node reduction  
→

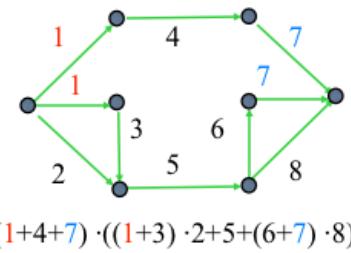


and the "dual"  
picture as in  
Dodd's algorithm

Measure 1: # node reductions



path expression  
→



$$(1+4+7) \cdot ((1+3) \cdot 2+5+(6+7) \cdot 8)$$

Measure 2: # duplicated subexpressions

Which measure is better?

- ▶ [Bein, Kamburowski & Stallman '92]
  - Measure 1 ≥ Measure 2
  - Measure 1 can be determined in polynomial time
- ▶ [Naumann '94]
  - Measure 1 = Measure 2

## Exercises

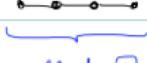
17.1 Consider series-parallel partial orders and their comparability graphs

An undirected graph is called a complement-reducible graph

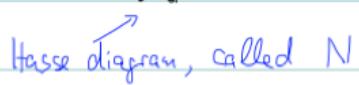
or simply cograph if every induced subgraph  $H$  of  $G$  has the property that  $H$  or its complementary graph  $\bar{H}$  is disconnected.

Show that  $G$  is a cograph iff it is the comparability graph of a series-parallel partial order

17.2\* Let  $G$  be a finite undirected graph. Show

$G$  is a cograph  $\Leftrightarrow G$  does not contain  as induced subgraph.  
called  $P_4$

17.3 Let  $G$  be a finite partial order. Show

$G$  is series-parallel  $\Leftrightarrow G$  does not contain  as induced suborder  
Hasse diagram, called  $N$

17.4 Show that Dodin's algorithm always terminates