Modeling the geometry of the Endoplasmic Reticulum network

Laurent Lemarchand
Reinhardt Euler

Congping Lin
Imogen Sparkes

Lab-STICC

ueb

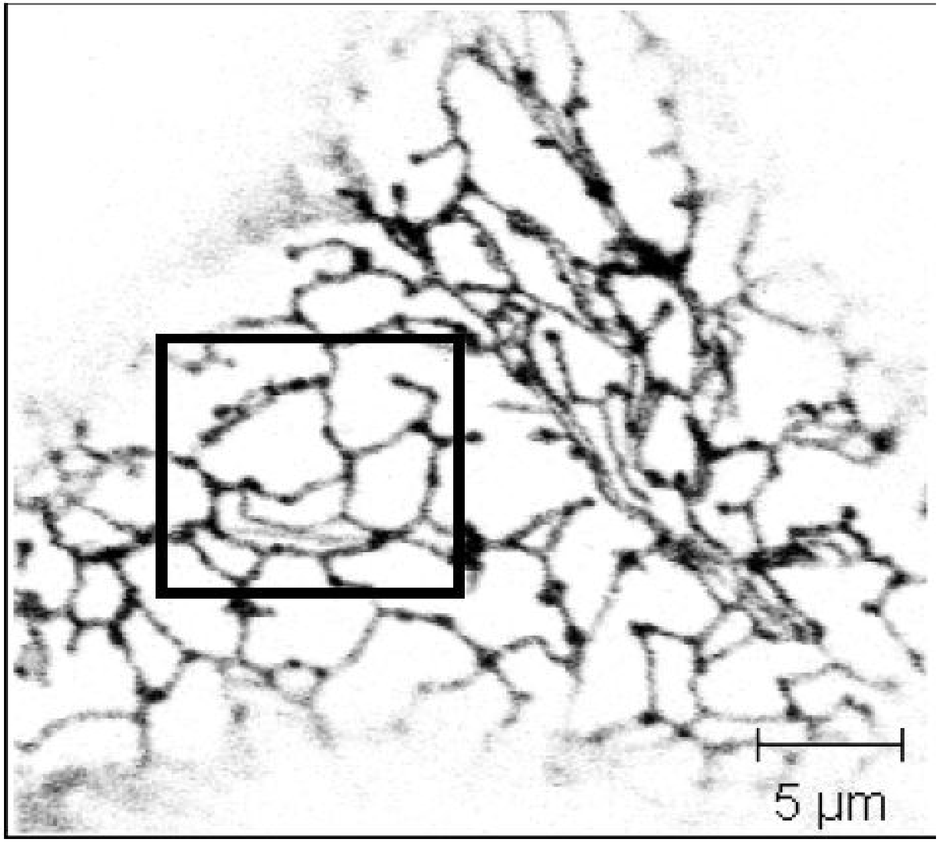
UBO
université de bretagne occidentale
Brest

UNIVERSITY OF EXETER

Endoplasmic Reticulum

- Membrane bound organelle
- Protein translocation, vesicle budding *[grand central station for biochemical compounds]*

- A network of tubules [edges] connected by nodes
  - persistent ones (anchoring to plasma membrane)
  - non-persistent ones (tubule junctions and tubule ends sensible to remodelling)



E.g tobacco leaf © American Society of Plant Biologists. www.plantcell.org

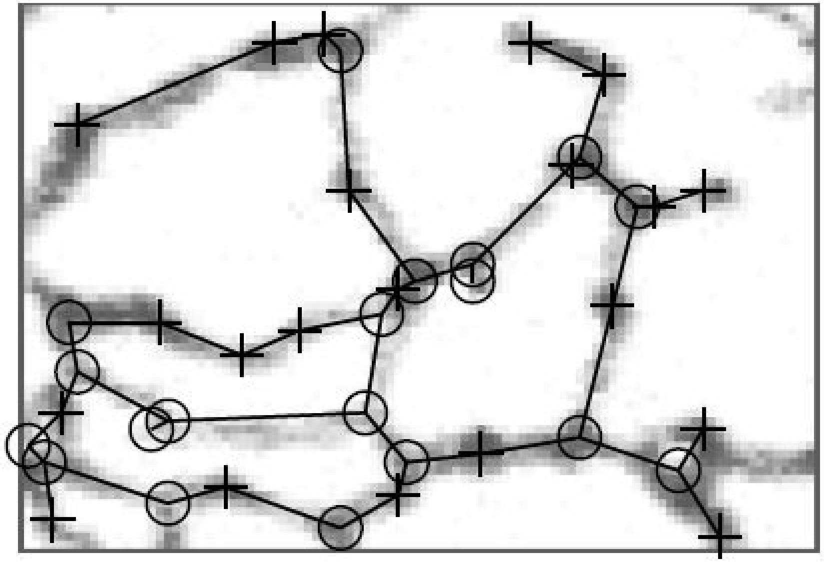
Understanding the ER dynamics

- Analysis of confocal microscopy data
- Is there and what is the optimization principle ?

Tubule interconnections

ER morphology in tobacco leaf epidermal cells

- Similar to Steiner trees or MST
- Additional cycles
- Branching nodes (3-way junctions) with angles around 120°



Abstracted geometric graphs

- From image processing :

O non-persistent nodes

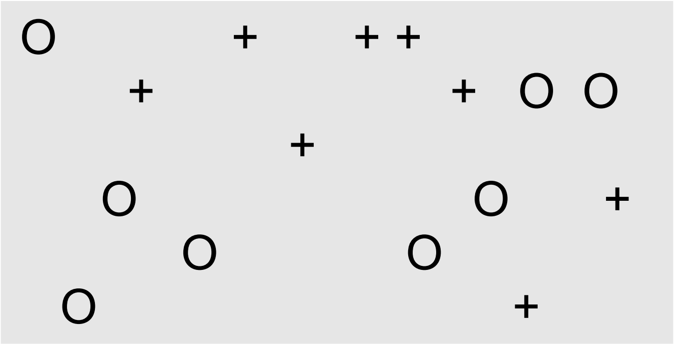
+ persistent nodes

Basic and full models

Compute a minimum length connected network

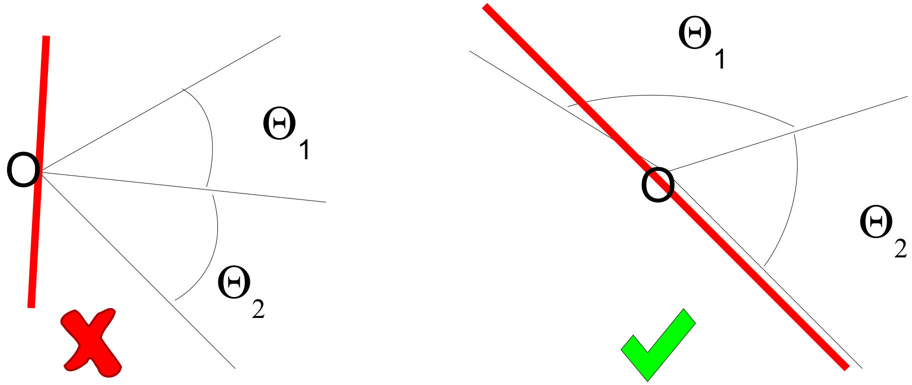
- Nodes (O/+) : tubule junctions extracted from images → V
- Detected branching nodes → $V_b \subseteq V$
- Plane graph

  [basic model]



- Angle constraint on branching nodes

  [full model]



Binary Problem Formulation

Given

- Weighted graph $G = (V, E, w)$
  - w_{uv} = euclidean distance between u and v
- $V_b \subseteq V$, a subset of branching nodes

$$\text{minimize} \sum_{x_{uv} \in E} w_{uv} x_{uv} \tag{1}$$

$$\text{subject to} \quad 1 \leq \sum_{v \neq u} x_{uv} \leq 3 \qquad \forall u \in V \setminus V_b, \tag{2}$$

$$\sum_{v \neq u} x_{uv} = 3 \qquad \forall u \in V_b, \tag{3}$$

$$\delta(W) \geq 1 \qquad \forall W \subset V, |W| \geq 2 \tag{4}$$

$$x_{uv} \in \{0, 1\}, \qquad \forall uv \in E \tag{5}$$

$$x_e + x_f \leq 1 \qquad \forall e, f \in E, e, f \text{ cross} \tag{6}$$

Binary Problem Resolution

Initial problem *BP*

- Eq. families (4) and (6) not included at beginning
  - Connected & plane result not guaranteed
- Need to exclude infeasible solutions

$$\text{minimize} \sum_{x_{uv} \in E} w_{uv} x_{uv} \quad (1)$$

$$\text{subject to} \quad 1 \leq \sum_{v \neq u} x_{uv} \leq 3 \qquad \forall u \in V \setminus V_b, \quad (2)$$

$$\sum_{v \neq u} x_{uv} = 3 \qquad \forall u \in V_b, \quad (3)$$

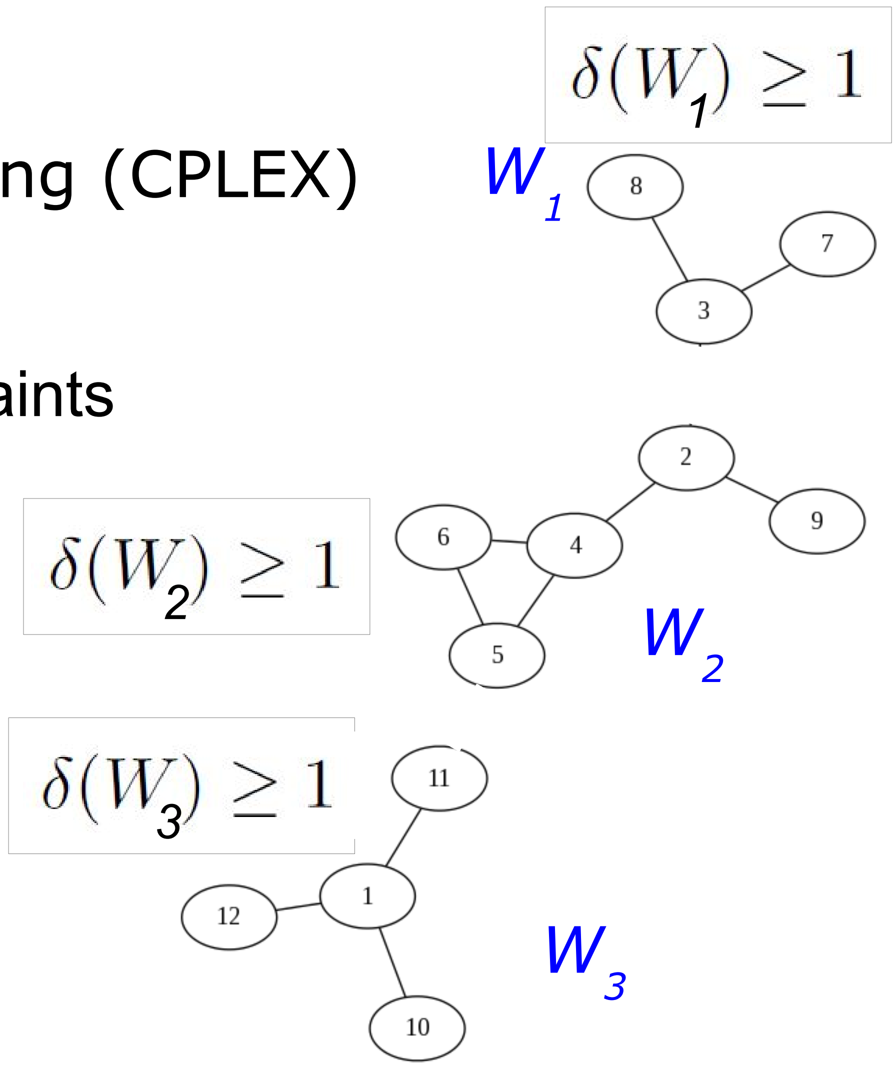
$$\delta(W) \geq 1 \qquad \forall W \subset V, |W| \geq 2 \quad (4)$$

$$x_{uv} \in \{0, 1\}, \qquad \forall uv \in E \quad (5)$$

$$x_e + x_f \leq 1 \qquad \forall e, f \in E, e, f \text{ cross} \quad (6)$$

Binary Problem resolution

- Solve *BP* by Integer Programming (CPLEX)
- Detect connected components
  Detect crossing edges
- Add corresponding (4) or (6) constraints



$$\delta(W) = \sum_{u,v} x_{uv}$$

$$u, v \in V, u \in W, v \notin W$$

- [Full problem] compute also angle compatibilities

LP-Relaxation resolution

Don't use built-in IP-tool branch & cut techniques

- Solve *RP* (relaxed version of *BP*), by *LP* tool (CPLEX)
- 0-1 constraints (5) replaced by

$$0 \leq x_{uv} \leq 1 \ \forall uv \in E$$

- **Branch & cut procedure**
  - Pre-solve version : swap to IP solver when violated cutting plane not found at the root node
  - Full branch & cut version : search for violated cutting plane for each node of the branch & cut tree

- Search for suitable cutting planes
  - Separation procedures

Cutting planes and separation procedures (1/3)

- Mincut based – equations (4)

$$\delta(W) \geq 1$$

  - BP : separation on connected components
  - Generalization for LP

- Separation procedure ($\rightarrow L_2$)
  - Find a min-cut (V_1, V_2) of V
  - If cut value ≤ 1, add inequality

$$\delta(V_1) \geq 1$$

Cutting planes and separation procedures (2/3)

- Multicuts based on partition of nodes [Cornuéjols et al. 1985, Grötschel et al. 1990)]

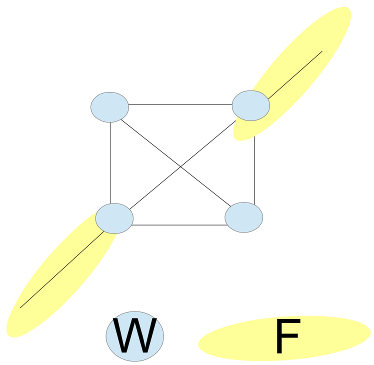
$$P = \{V_1, V_2, ..., V_p\} \qquad \frac{1}{2}\sum \delta(V_i) \geq (p-1)$$

- 2 different separation procedures
  - Recursive cuts [Barahona 2000] ($\rightarrow L_r$)
    - Given a k-partition, compute k internal mincuts and find the minimal cut value. Split the corresponding set
    - Check inequality and stop if violated [**found**]
    - Stop when k = p [**failed**]
  - Parametric cuts ($\alpha \in]0..1]$)($\rightarrow L_k$)
    - Build $G' = (V, E' = \{e \in E | x_e \geq \alpha\})$
    - Components of G' define the partition

Cutting planes and separation procedures (3/3)

- Blossom inequalities based on Edmonds' description of b-matching polytope

$$\sum_{e \in E(W)} x_e + \sum_{f \in F} x_f \leq \left\lfloor \frac{3|W| + |F|}{2} \right\rfloor, \forall W \subset V, F \subset \delta(W) \quad \text{with} \quad 3|W| + |F| \text{ odd}$$



- Separation procedure [Letchford et al. 2008] ($\rightarrow L_b$)
  - Based on cut trees (Gomory-Hu)
  - Special blossom procedure for cubic case ($V_b = V$) [Letchford et al. 2004]

Numerical tests

- Runtimes for solving testcases
  - Basic model
  - Full model

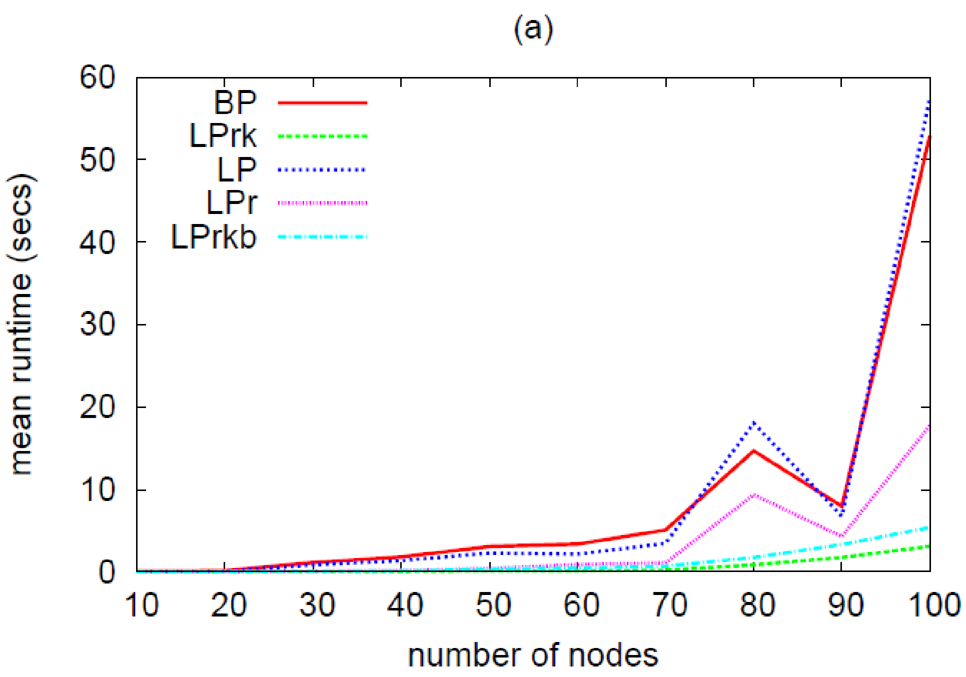
  - Binary problem solution procedure BP
  - LP, LP_r, LP_{rk}, LP_{rkb}, on root node only [pre-solve phase]
  - LP, LP_r, LP_{rk}, LP_{rkb}, with full branch & cut

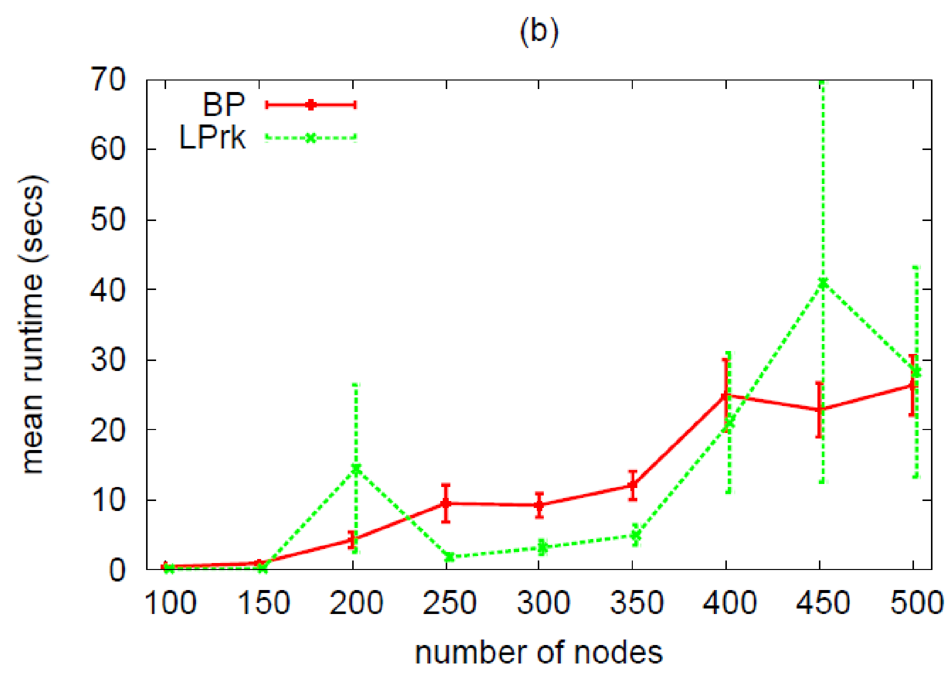
  - Random testcases (size, percentage of branching nodes)
  - Real-life testcases (50 frames, $n \approx 70$, $|V_b| \approx 40\%$)

- Comparison of computed and actual topologies
  - Real-life testcases only
  - Basic and full models
  - Comparison metrics

Runtimes
BP *vs* LP_x , pre-solve only
random testcases, basic model

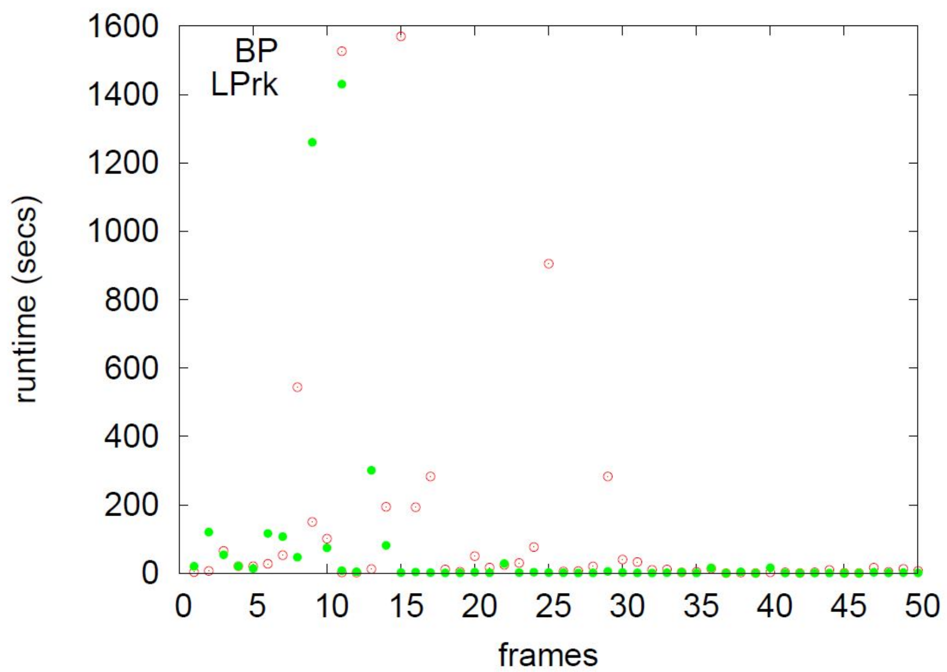
- Cutting techniques LP_{rk} and LP_{rkb} are the most efficient (fig. a)
- LP_{rkb} (blossom cuts) costly as compared to LP_{rk}
- Good behavior of LP_{rk} for large cases (fig. b)





Runtimes
BP *vs* LP_x , pre-solve only
Real-life testcases, full model

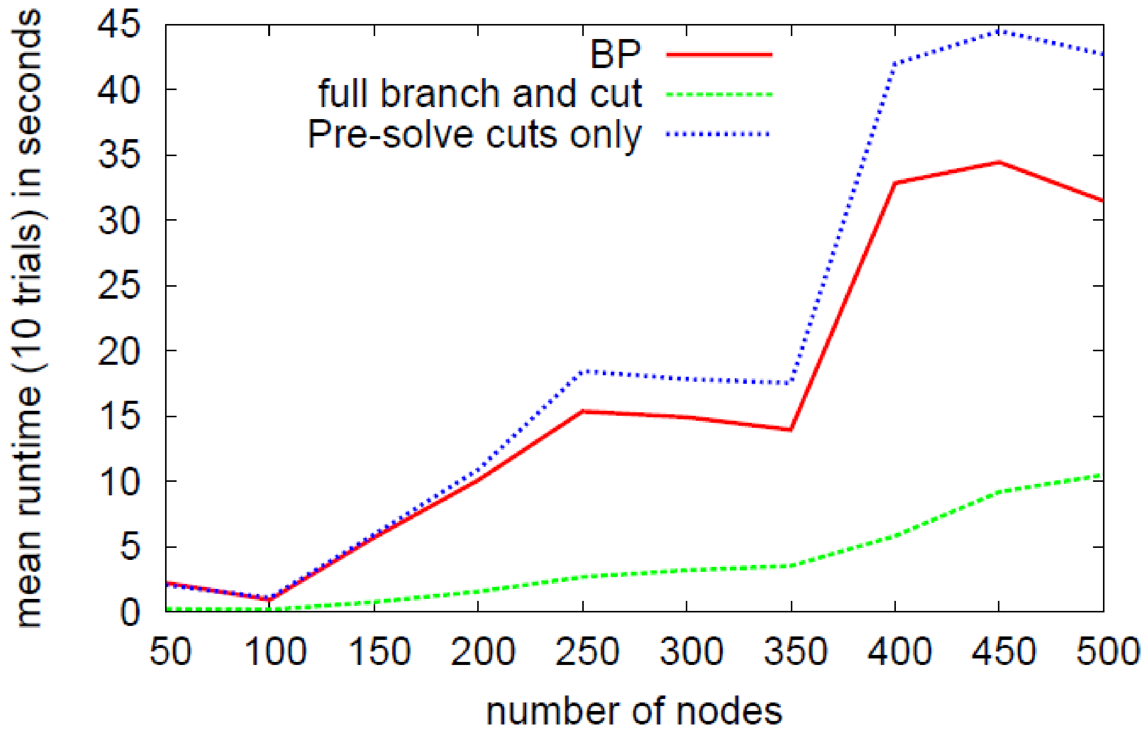
- Cutting techniques LP_{rk} and LP_{rkb} are the most efficient for our 50 testcases
- LP_{rkb} (blossom cuts) costly as compared to LP_{rk}
- Good behavior of LP_{rk} for full model (planarity and angle checks)



Runtimes
BP *vs* LP_x , full branch & cut
random testcases, basic model

Latest results

- Connected component cuts are efficient for inner search nodes
- Computing LP_x cuts at each inner node is very costly (→ removed)
- Good behavior for large instances (time divided by 3)

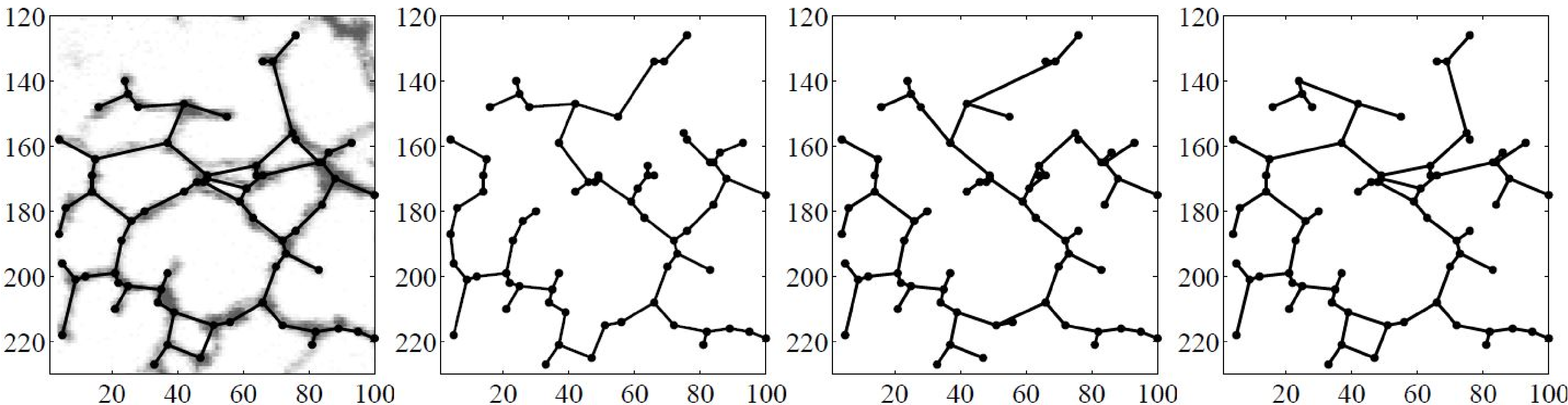


Quality of computed networks
50 real-life testcases

- High level similarities (similarity → 1, matching → 0)
- Well adapted except a few missing cycles

Computed network G_1

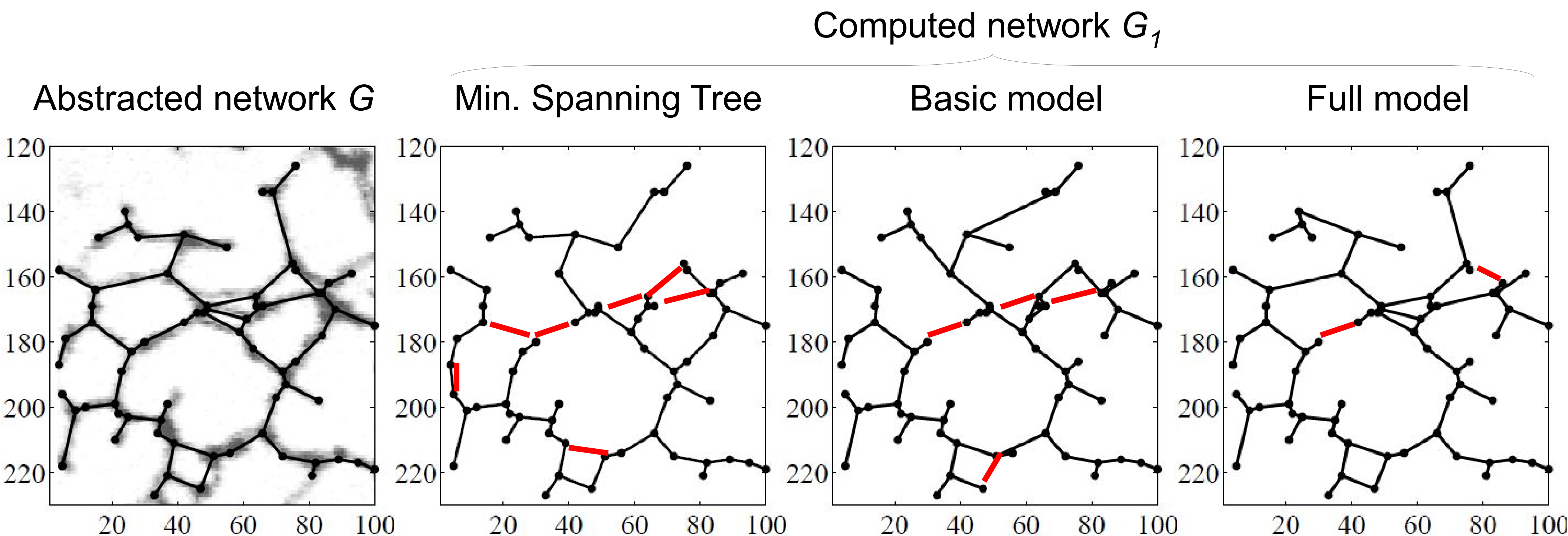
Abstract network G | Min. Spanning Tree | Basic model | Full model



N=50	MST model	basic model	full model ($\theta = 160^o$)	full model ($\theta = 170^o$)	full model ($\theta = 180^o$)
$s(G, G_1)$	0.7376 ± 0.0015	0.9065 ± 0.0103	0.9185 ± 0.0092	0.9270 ± 0.0081	0.9271 ± 0.0084
$m(G, G_1)$	0.2930 ± 0.0195	0.0106 ± 0.0011	0.0096 ± 0.0011	0.0087 ± 0.0010	0.0087 ± 0.0011

Quality of computed networks
50 real-life testcases

- High level similarities (similarity → 1, matching → 0)
- Well adapted but without cycles



N=50	MST model	basic model	full model ($\theta = 160^o$)	full model ($\theta = 170^o$)	full model ($\theta = 180^o$)
$s(G, G_1)$	0.7376 ± 0.0015	0.9065 ± 0.0103	0.9185 ± 0.0092	0.9270 ± 0.0081	0.9271 ± 0.0084
$m(G, G_1)$	0.2930 ± 0.0195	0.0106 ± 0.0011	0.0096 ± 0.0011	0.0087 ± 0.0010	0.0087 ± 0.0011

Conclusion and future work

- Good results in terms of similarities for computed networks

- Cutting techniques efficient, but sometimes costly
  - Edmonds' blossom inequalities deceiving
  - B&C improves computing times by factor of 1/3
  - Find new cuts and separation procedures
- B&C features of CPLEX not totally exploited yet

- Cubic case $V_b = V \rightarrow \forall v \in V, d(v) = 3$ [TSP: d(v) = 2]
  - Separation procedure for blossom inequalities [Letchford et al. 2004] implemented
  - Current work is on description of new cutting planes