More shrinkage for the Multiterminal Cut problem

Dia-Powered Patatoïds™ included

Bram De Wachter, Alexandre Genon, Thierry Massart

{ **bdewacht, agenon, tmassart** }**@ulb.ac.be**

Outline

- Introduction and motivation
- Formal description
- Known solutions
- More shrinkage !
  - Using local heuristics
  - Using global heuristics, based on ST-cut
- Observations on ST-cut
- Results

Introduction and Motivation

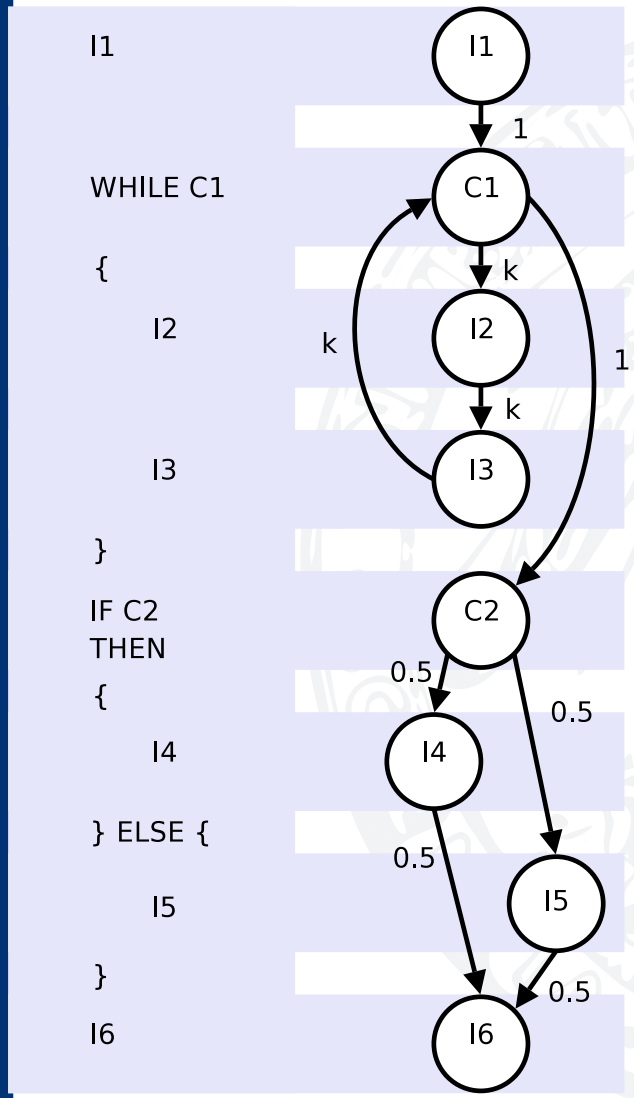
- Given
  - a program text (control flow)
  - and an estimate of the number of times each instruction is executed
- In dSL, the program must be distributed
  - Each instruction must be localized on a particular site $\in \{1, ..k\}$
  - Some instructions are fixed (because they access actuators/sensors that are fixed on a certain site)
- Question : assign each instruction to a site such that, during the execution, the control flow switches a minimal number of times between the different sites.

Example

```
I1


WHILE C1

{

    I2


    I3

}

IF C2
THEN
{

    I4

} ELSE {

    I5

}

I6
```

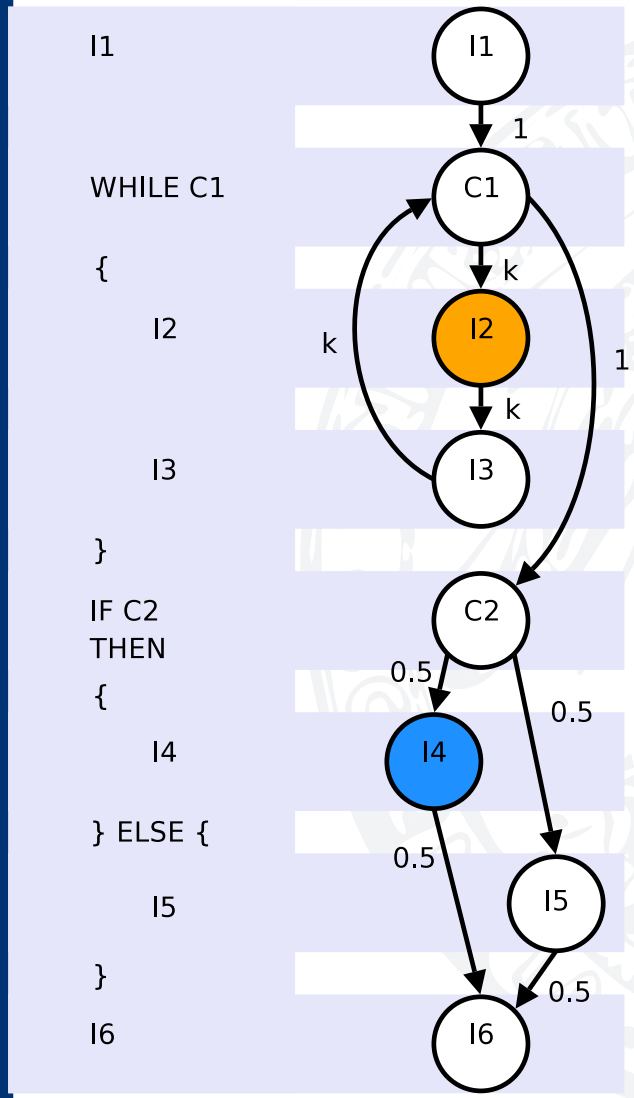
Given a program text

Example



Obtain a control flow graph weighted by a *realistic* estimate of the control flow's frequency during execution

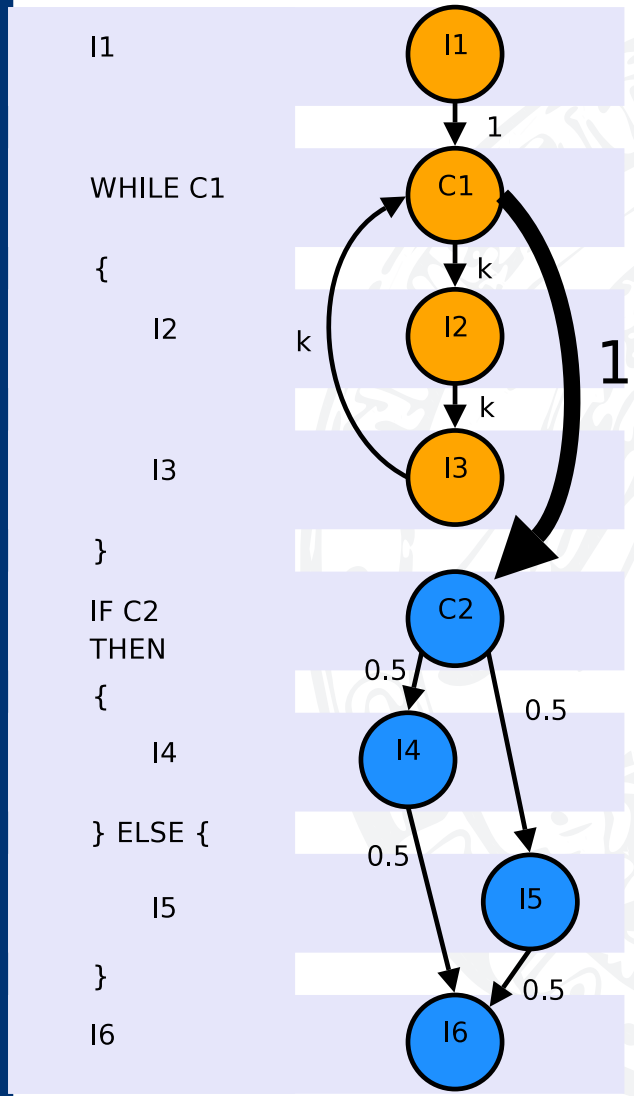
Example



- Some instructions (i_2 and i_4) are accessing variables that must be localized on a certain site. Which means that these instructions are fixed on those sites.
- The assignment of an instruction to a site, is expressed in terms of assigning the instruction a color.
- The problem is then to color all instructions such that

the sum of the weights of the edges between nodes of different colors is minimal

Example



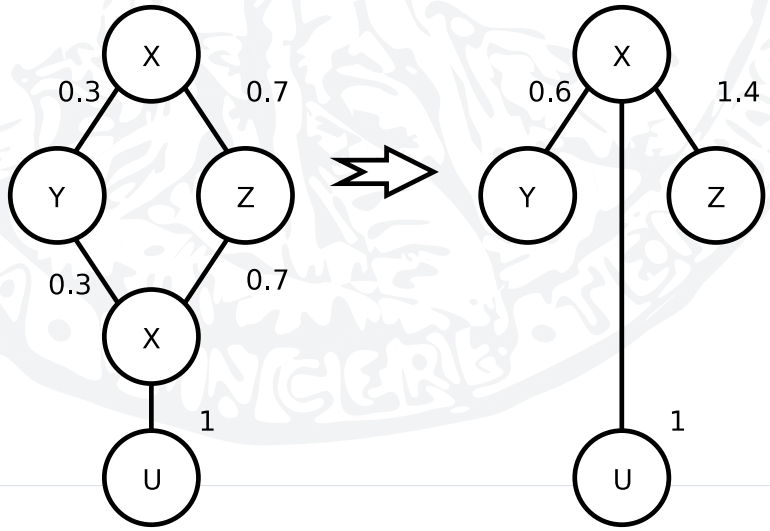
- An optimal solution in this example is of weight 1
- **Remark:** the weight is a sum, so the graph is undirected

Formal definition

- **Theorem:** The dSL-coloring problem is NP-Complete even if only a sequence of instructions is used (without `IF` and `WHILE`)
- Equivalent to the multiterminal cut problem

> Given a weighted, undirected graph $G(V, E, w)$, $E \subseteq V \times V$, $w : E \mapsto \mathbb{Z}^+$ and a set of terminals $T = \{s_1, ..., s_k\} \supseteq V$
> Partition V in $V_1, .., V_k$ such that
> $\sum_{\{(v_i,v_j)|v_i \in V_i, v_j \in V_j, i \neq j\}} w(v_i, v_j)$ is minimal

- Arbitrary graphs can be obtained by merging nodes that use the same variables



Formal definition (2)

The multiterminal cut problem ([DJP+94])

- is NP-Complete even with all weights = 1
- is polynomially solvable if $k = 2$ known as ST-Cut
  Famous min-cut max/flow theorem by *Ford/Fulkerson*
  best algorithm (Goldberg/Tarjan) solves in $nm \log(\frac{n^2}{m})$ with $n = |V|, m = |E|$
- is polynomially solvable for planar graphs

Known solutions

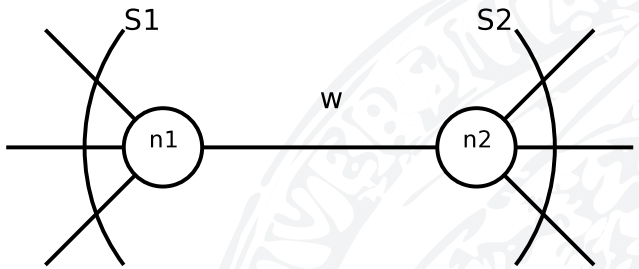
- Two approaches
  - Dalhaus et al
    - Based on min-cut/max-flow
    - $\alpha = 2 - \frac{2}{k}$
    - Isolation heuristics, find an isolating cut for each terminal, take union of $k-1$ such cuts
    - $O(knm \log(n^2/m))$
  - Calinescu et al
    - Based on Linear Programming
    - $\alpha = 1.5 - \frac{1}{k}$
- But what about the possibility to reduce the instance size ?
  - Transform G into G' ($G > G'$) such that any optimal cut in G' is an optimal cut in G
  - Orthogonal with existing algorithms

Shrinkage for the Multiterminal Cut

- Shrinkage obtained by
  - Random sampling [Kar94]
  - Local heuristics [HK01]
    - *Dominant edge*
    - *Zeroing*
    - *Articulation point*
    - *Independent subgraphs*
    - *Machine Cut*
  - Isolation heuristics [DJP+94]
- More shrinkage using
  - multiple ST-Cuts, resulting in a generalization for
    - *Dominant edge*
    - *Machine Cut*
    - *Independent subgraphs*
    - *Isolation heuristics*
  - specific reduction criteria, resulting in generalizations for
    - *Dominant edge*
    - *Articulation point*

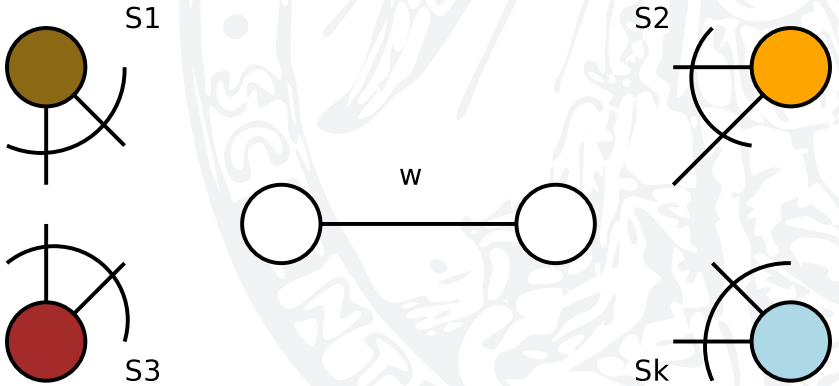
Local heuristics [HK01]

- Dominant edge



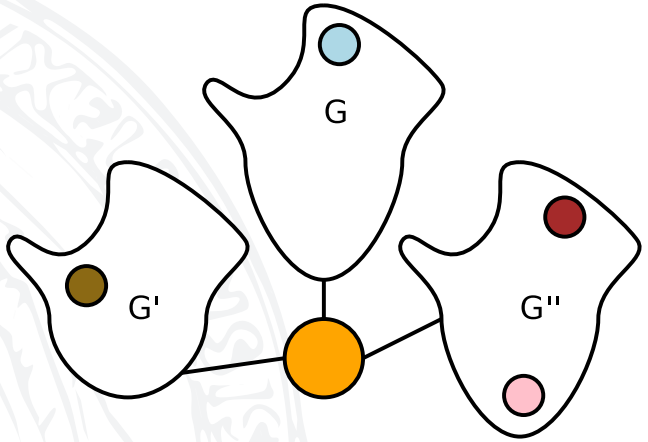
$$w \geq S_1 \vee w \geq S_2$$

- Machine cut

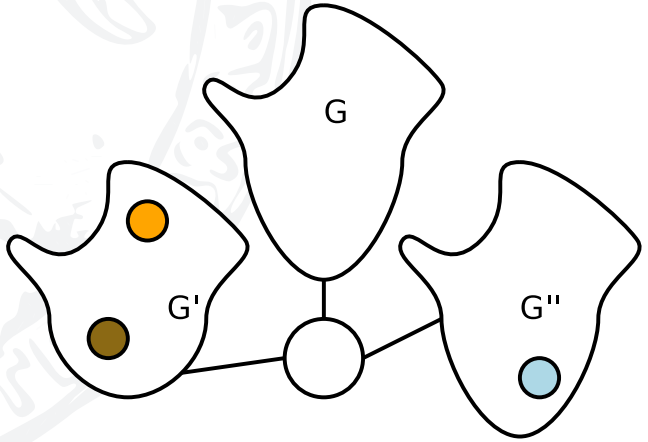


$$S_1 \geq S_2 \geq ... \geq S_k \wedge w \geq S_2$$

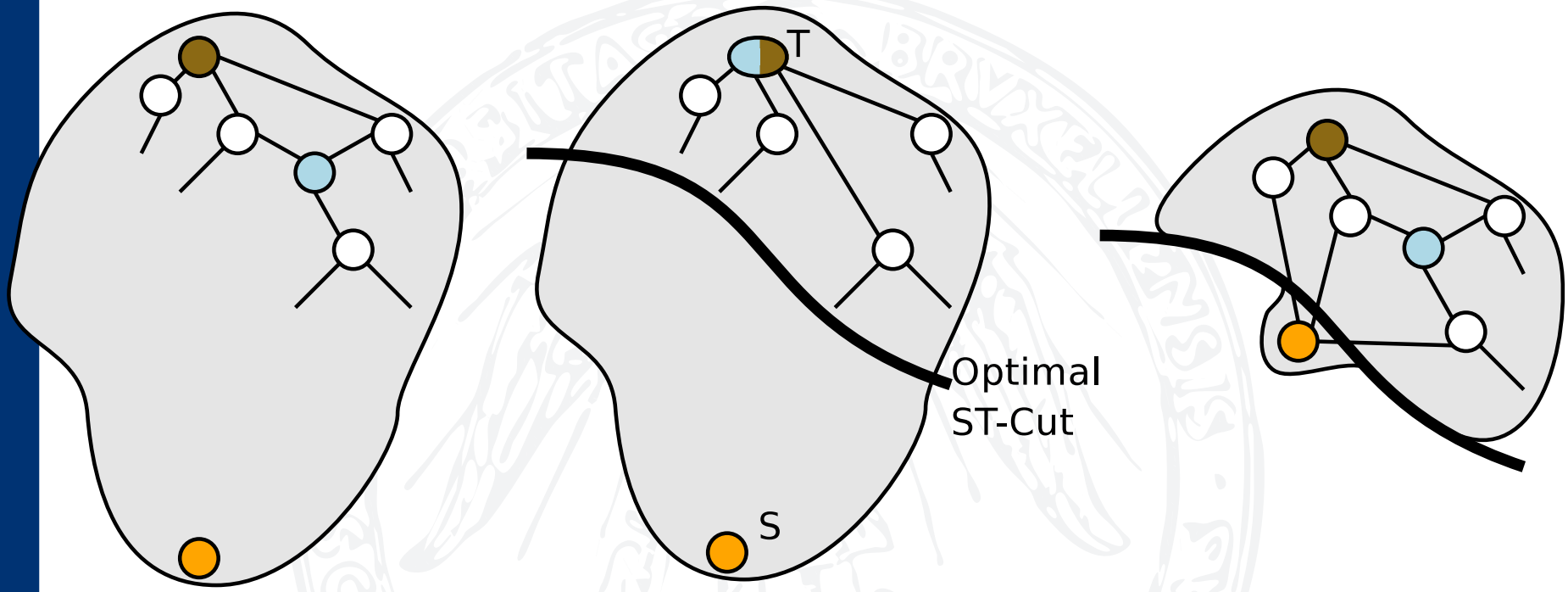
- Independent subgraph



- Articulation point



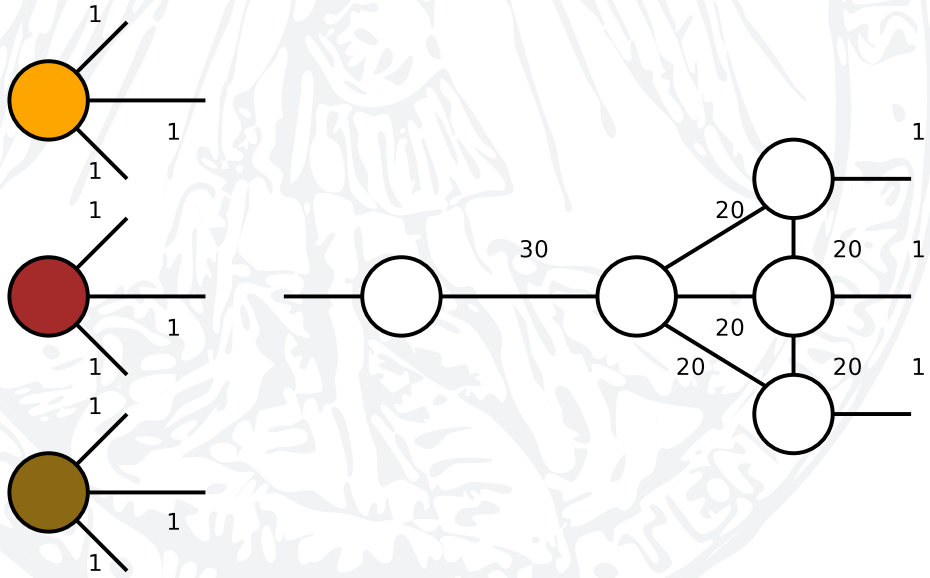
Isolation heuristics [DJP$^+$94]



Theorem 1: Let G'_i be the graph where all terminals in $T \setminus \{s_i\}$ are merged into t, and $(C, \overline{C})$ the min-ST-Cut between s_i and t, then there exists an optimal k-cut $(V_1, ..., V_k)$ of G such that all nodes in C are in one of the V_j.

A generalized global heuristics

- **Theorem 2:** Given a weighted graph $G(V, E, w)$ and $n \in V$, Let G'_i be the graph where all terminals in $T \setminus \{n\}$ are merged into t, and $(C, \overline{C})$ the min-ST-Cut between n and t then there exists an optimal k-cut of G $(V_1, ..., V_k)$ such that all nodes in C are in one of the V_j.
- This handles all cases of *dominant edge*, *machine cut*, *independent subgraphs*, *isolation heuristics* and additional cases :



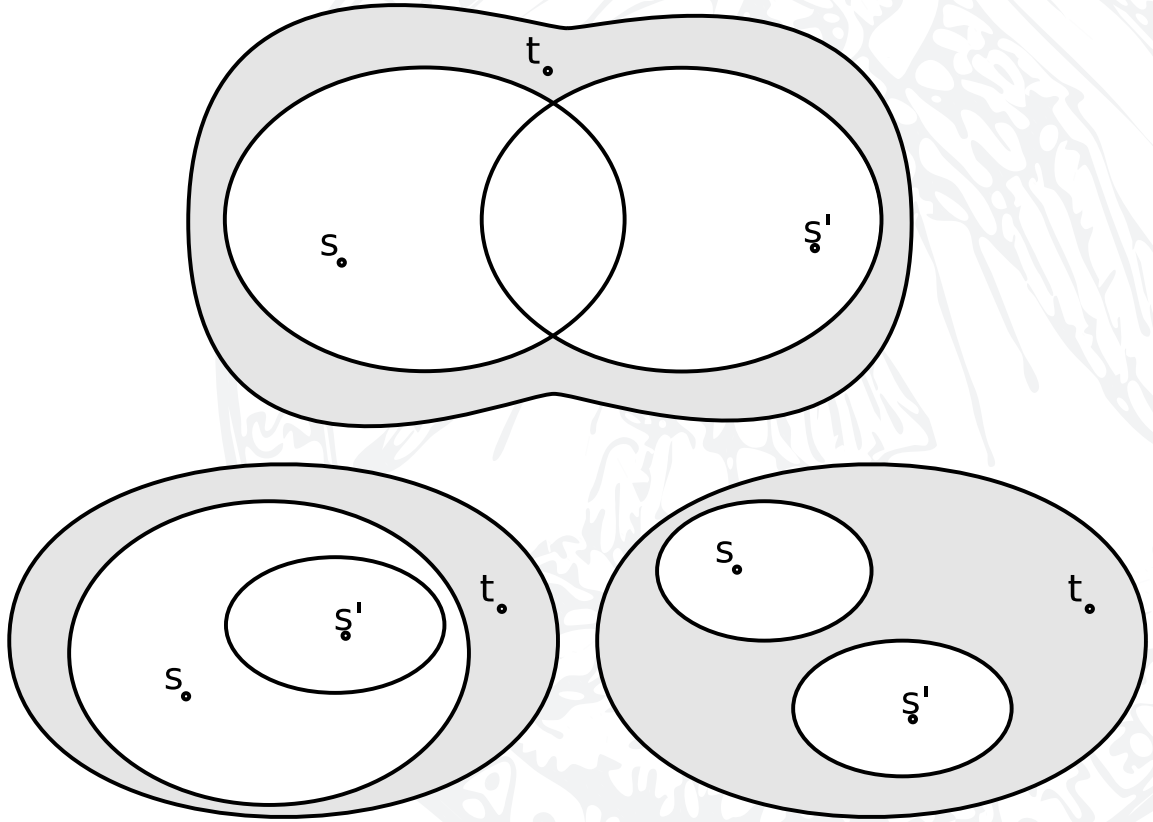
- Requires a priori n ST-cuts $\Rightarrow O(n \cdot nm \log(\frac{n^2}{m}))$

Some observations on ST-Cut *(1)*

- Given a weighted graph $G(V, E, w)$ and three different vertices $s, s', t \in V$. Define min-ST(s, t) as the set of cuts separating s and t with minimal weight. We define the set max-min-ST(s, t) as the set of nodes left connected to s by the cut $(C, \overline{C}) \in$ min-ST(s, t) such that $|C|$ is maximal.
- **Theorem 3:** Given a weighted graph $G = (V, E, w)$ and two vertices $s, t \in V$. We have if $C_1 =$max-min-ST(s, t) and $C_2 =$max-min-ST(s, t), we must have $C_1 = C_2$.
  I.e. there is only one maximal size minimal ST cut.

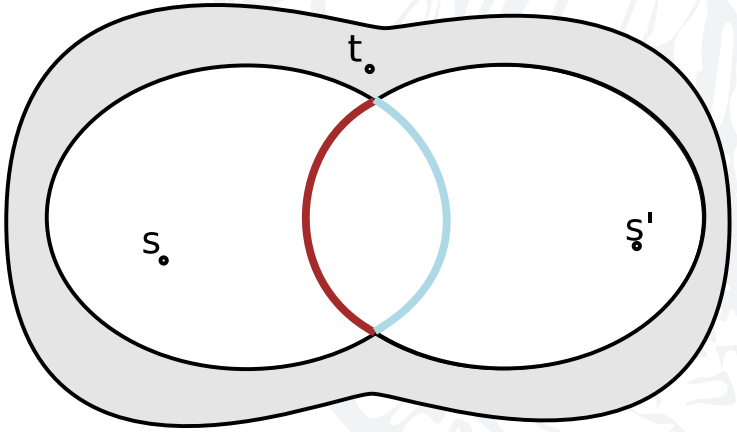
Some observations on ST-Cut *(2)*

- **Theorem 4:** For any three vertices s, s', t of V, we have the following property : if $s' \in$ max-min-ST(s,t), then max-min-ST$(s',t) \subseteq$ max-min-ST(s,t).



Some observations on ST-Cut *(2)*

- **Theorem 4:** For any three vertices s, s', t of V, we have the following property : if $s' \in$ max-min-ST(s,t), then max-min-ST$(s',t) \subseteq$ max-min-ST(s,t).
- **Theorem 5:** Given a weighted graph $G(V,E,w)$ and three different vertices $s, s', t \in V$. Let $S =$max-min-ST(s,t), and $S' =$max-min-ST(s',t). If $I = S \cap S' \neq \phi \wedge S \neq I \wedge S' \neq I$, then $w(I, S \setminus I) = w(I, S' \setminus I)$.



Some observations on ST-Cut *(2)*

- **Theorem 4:** For any three vertices s, s', t of V, we have the following property : if $s' \in$ max-min-ST(s,t), then max-min-ST$(s',t) \subseteq$ max-min-ST(s,t).
- **Theorem 5:** Given a weighted graph $G(V,E,w)$ and three different vertices $s, s', t \in V$. Let $S =$max-min-ST(s,t), and $S' =$max-min-ST(s',t). If $I = S \cap S' \neq \phi \wedge S \neq I \wedge S' \neq I$, then $w(I, S \setminus I) = w(I, S' \setminus I)$.

☺ Theorems **3-5** lower the complexity since

- We can contract edges in any order
- We can contract edges on the fly
- Recontraction is unnecessary

☹ However, worst case stays $O(n^2 m \log(\frac{n^2}{m}))$ (?)

A local unshackling heuristics

- A procedure to solve the multiterminal cut:

```
reduce(); while(non_terminals exist) {
    unshackle(); // Contract 1 edge
    reduce();
}
```

- **Theorem 6:** If G is irreducable ($\forall n \in V$: max-min-ST$(n, t^*) = \{n\}$), and G' is obtained after contracting edge $\{n', n''\} \in E$, then in G' we have that $\forall n \in V \setminus \{n', n''\}$: max-min-ST$(n, t^*) = \{n\}$

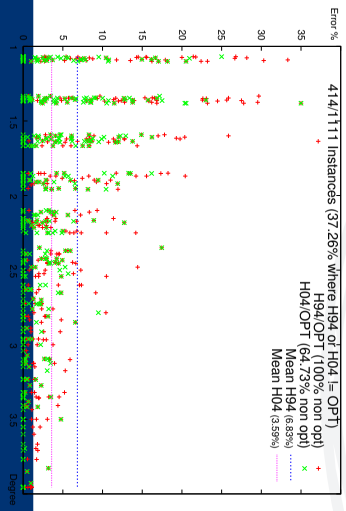
  $\Longrightarrow$ Worst case $O(n^2 m \log(\frac{n^2}{m}))$ if `unshackle()` easy

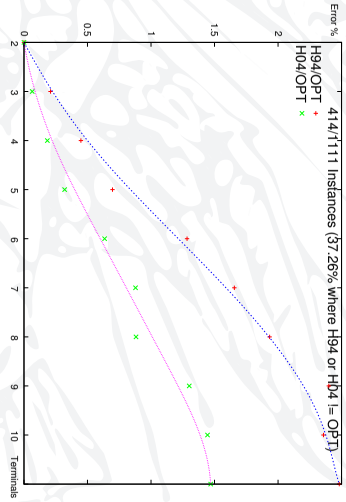
- Find a good local `unshackle()` procedure
  - Greedy: Take heaviest edge
  - Error-reduction: Take edge with lowest error bound
  - Mistery: Take edge $\{n_1, n_2\}$ that maximizes $\sum_{(n',n_1)} w(n', n_1) + \sum_{(n',n_2)} w(n', n_2)$

- Loss of $\alpha = 2 - \frac{2}{k}$ approximation bounds

Results

- Using the Mistery `unshackle()` procedure has best results
- Random graphs

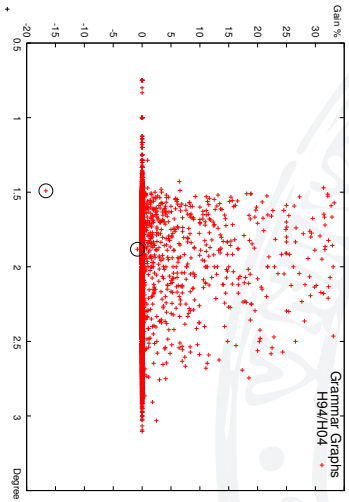




- No cases where we did worse than [DJP+94]

Results (2)

- Grammar graphs with merged nodes



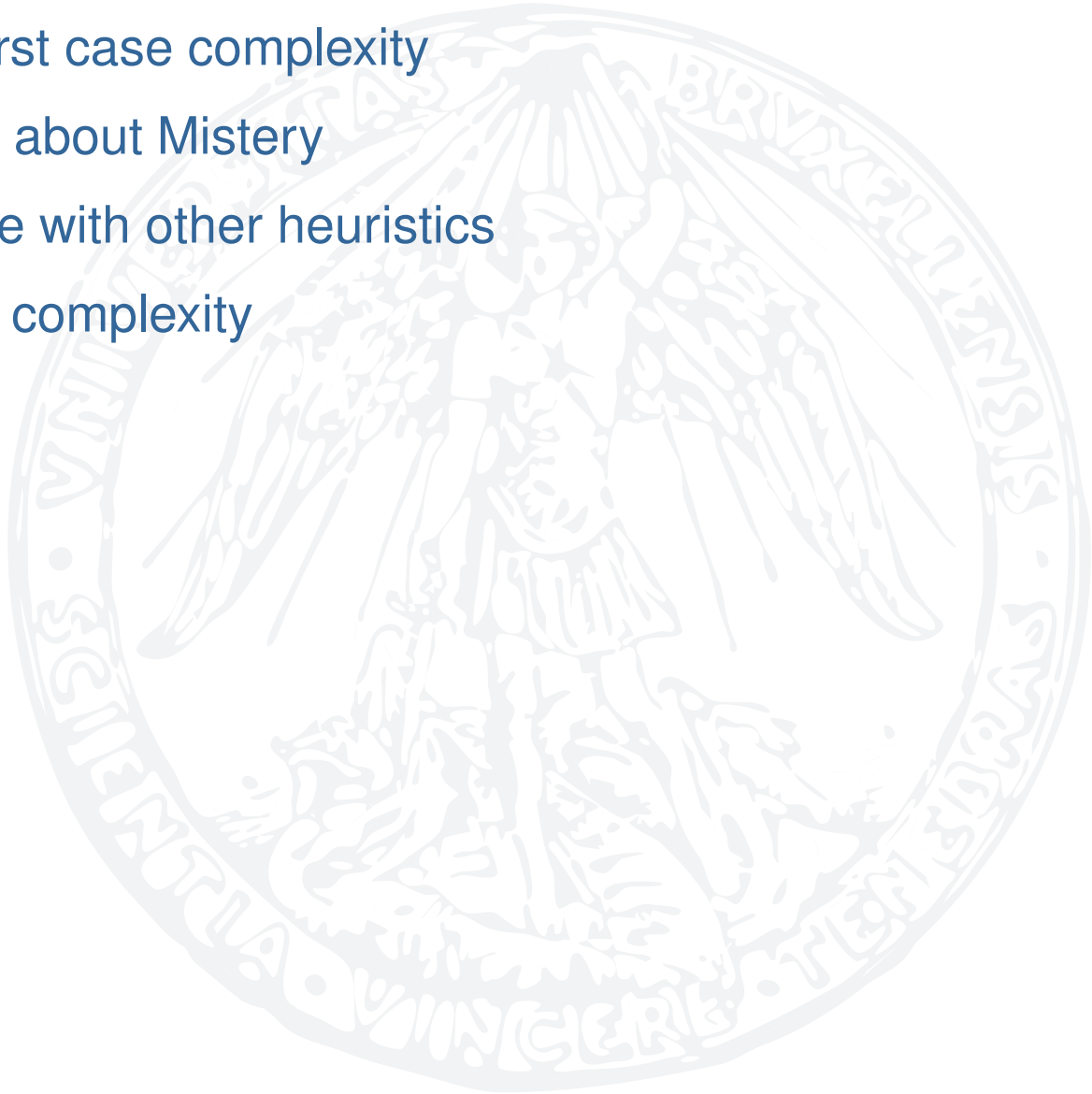
Mean error ($25k$ graphs) between optimal and

- [DJP+94] 0.369% (4.11% non opt)
- Mistery 0.299% (1.42% non opt)

- Only 2 cases where we did worse

Future works

- Find worst case complexity
- Find out about Mistery
- Compare with other heuristics
- Improve complexity



Bibliography

[DJP+94] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, P. D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[HK01] K. Hogstedt and D. Kimelman. Graph cutting algorithms for distributed applications partitioning. *SIGMETRICS Performance Evaluation Review*, 28(4):27–29, 2001.

[Kar94] David R. Karger. Random sampling in cut, flow, and network design problems. In *Proc. 26 th Annual ACM Symposium on the Theory of Computing*, pages 648–657, 1994.