Redundant constraints in the standard formulation for the clique partitioning problem

Noriyoshi Sukegawa Atsushi Miyauchi

Tokyo Institute of Technology

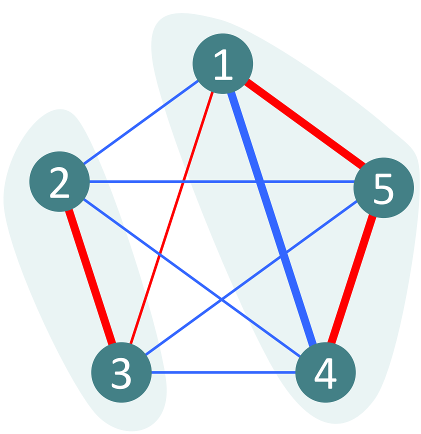
EURO | INFORMS MMXIII,

MC-28: Geometric Clustering 2 in stream Geometric Clustering

1st July 2013

The clique partitioning problem (CPP)

- Given an complete and undirected graph $G = (V, E)$, with an edge weights $w : E \to \mathbb{R}$, CPP is to find
  - a partition $\{V_1, V_2, \ldots, V_l\}$ of V maximizing
  - the total weight of edges within the components.



- # components = 2
- objective value = 3*(+2) + (-1) = +5

(edges in red : +2, edges in blue : -1)

Some features

- Different from other clustering problems,
  - we have to deal with negative edge weights, and
  - # components is not fixed.

- In general, CPP is NP-hard [Wa ’86].

- Applications: aggregation problem, flight-gate scheduling, group technology problem, modularity maximization … .

Standard formulation

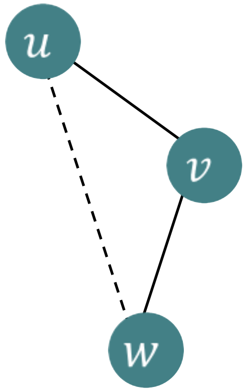
- Assign 0-1 variables for the edges [GrWa '89].

$$x(e) = \begin{cases} 1 & (\text{ if } u, v \in V_i \text{ for some } i\ ) \\ 0 & (\qquad \text{otherwise} \qquad) \end{cases} \qquad (e = \{u, v\} \in E)$$

- The constraints are simple:
  - $x(u,v) = 1,\ x(v,w) = 1 \Rightarrow x(u,w) = 1.$

$$\equiv\ x(u,v) + x(v,w) - x(u,w)\ \leq 1$$

  - referred to as the transitivity constraints.



Transitivity constraints

- Note that # transitivity constraints is $O(n^3)$.
  - Even when $n = 300$, it amounts to about 13 million !!
  - This size issue limits the application of CPP.

- **Question**: Do we have to consider all of them?
  - To describe the feasible region, of course, yes.
  - However, to solve the problem, some of them would be ``redundant'', namely no.

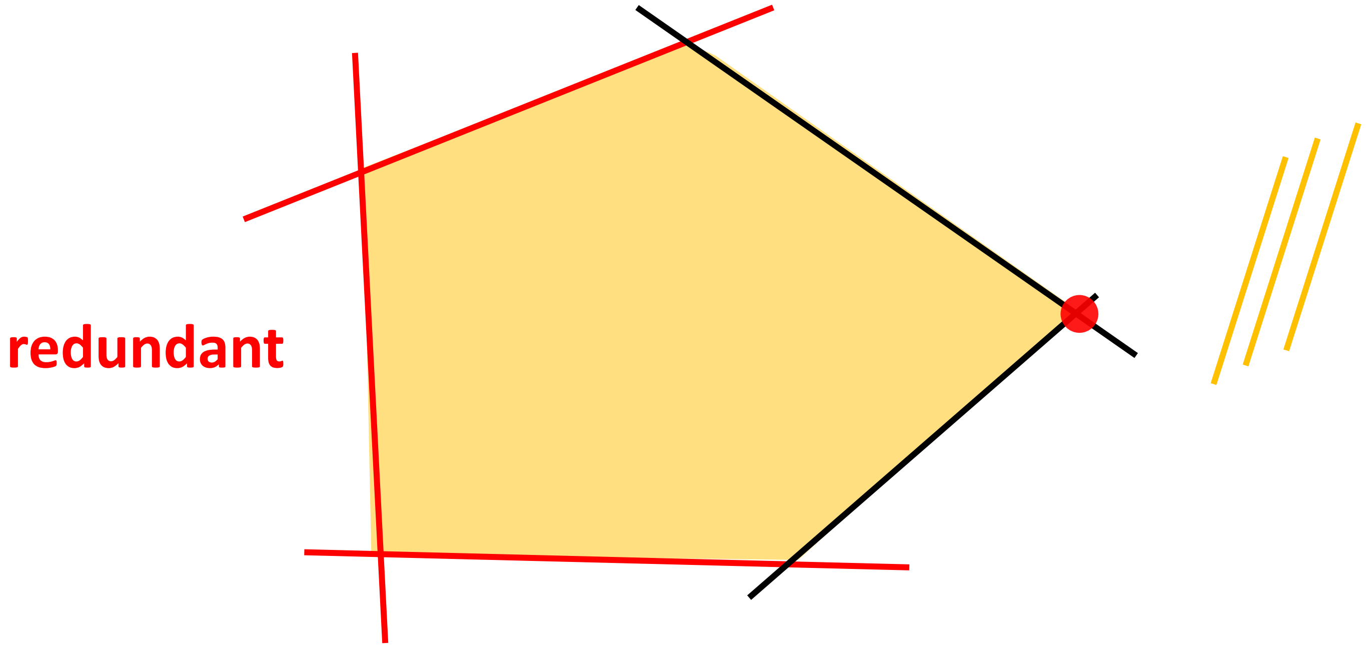
Our result

- **Theorem**: A set of the transitivity constraints $x(e) + x(f) - x(g) \leq 1$ such that the first two edge weights (say $c(e), c(f)$) are negative is redundant.

- **Definition**: We call a set of constraints redundant if the optimal solution set remains unchanged even if we delete them.

Our result

- **Theorem**: A set of the transitivity constraints $x(e) + x(f) - x(g) \leq 1$ such that the first two edge weights (say $c(e), c(f)$) are negative is redundant.



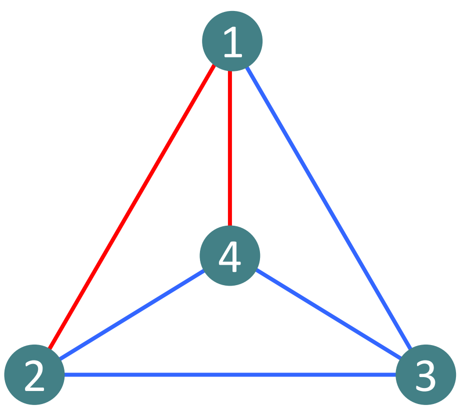
Our result

- **Theorem**: A set of the transitivity constraints $x(e) + x(f) - x(g) \leq 1$ such that the first two edge weights (say $c(e), c(f)$) are negative is redundant. **(RC)**

- **Definition**: We call a set of constraints redundant if the optimal solution set remains unchanged even if we delete them.

Our result

- **Theorem**: A set of the transitivity constraints $x(e) + x(f) - x(g) \leq 1$ such that the first two edge weights (say $c(e), c(f)$) are negative is redundant. **(RC)**



$$\begin{cases} x(1,3) + x(3,4) - x(1,4) \leq 1 \\ x(2,4) + x(4,3) - x(2,3) \leq 1 \\ x(4,3) + x(3,2) - x(4,2) \leq 1 \\ x(3,2) + x(2,4) - x(3,4) \leq 1 \end{cases}$$

$\Rightarrow$ satisfy **(RC)**

Background

- We are motivated by a result in Dinh and Thai (2011) who
  - dealt with the Modularity maximization ($\subseteq$ CPP)
  - and showed the redundant constraints.

  (This is not their main result but just a part of their study.)

- By carefully reading, we see that their proof can be generalized to CPP, easily.
- Through this, we can improve their result very slightly.

Preparation

- First, let's observe the optimal solution x^* to a constraint relaxation problem of CPP, denoted by CPP(M).

$\forall M \subseteq$ (all the constrains)

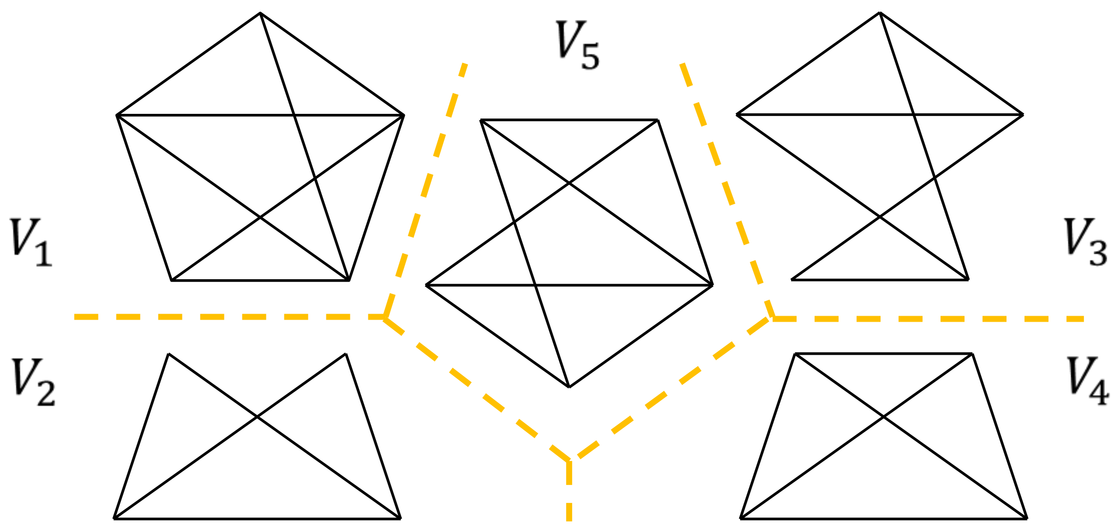
$$
\begin{aligned}
\textbf{max.} \quad & \sum_{e \in E} c(e)x(e) \\
\textbf{s.t.} \quad & x(e) + x(f) - x(g) \leq 1 && (\forall (e,f,g) \in M) \\
& x(e) \in \{0,1\} && (\forall e \in E)
\end{aligned}
$$

- Since, we delete some transitivity constraints, x* may be infeasible to the original CPP.

Trivial lemma

- $E^* = \{e \in E : x^*(e) = 1\}$: the set of edges corresponds to x^*.

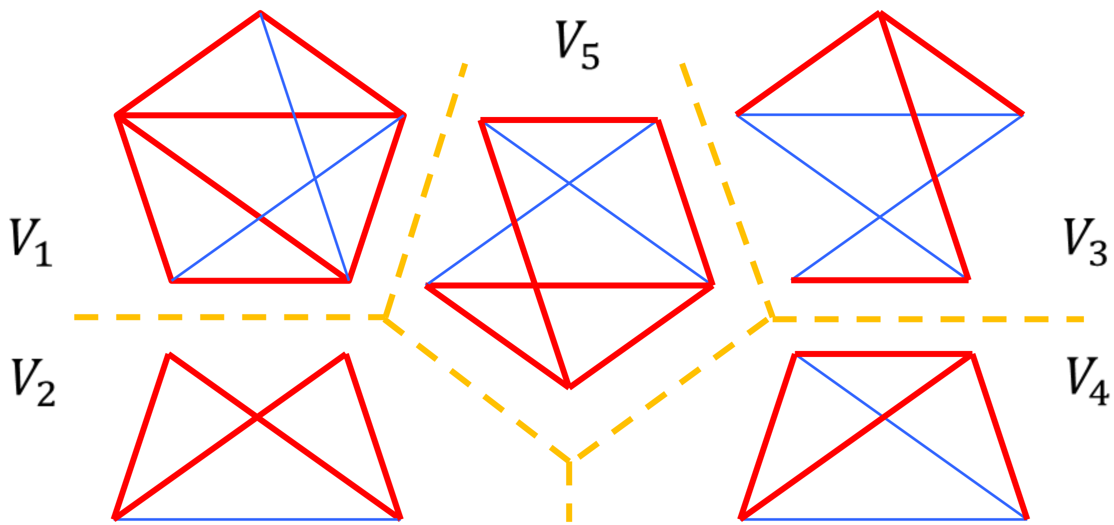
- **Lemma**: Each component is connected with edges with nonnegative weights (Intuitively clear).



Trivial lemma

- $E^* = \{e \in E : x^*(e) = 1\}$: the set of edges corresponds to x^*.

- **Lemma**: Each component is connected with edges with nonnegative weights (Intuitively clear).



Trivial lemma

- $E^* = \{e \in E : x^*(e) = 1\}$: the set of edges corresponds to x^*.

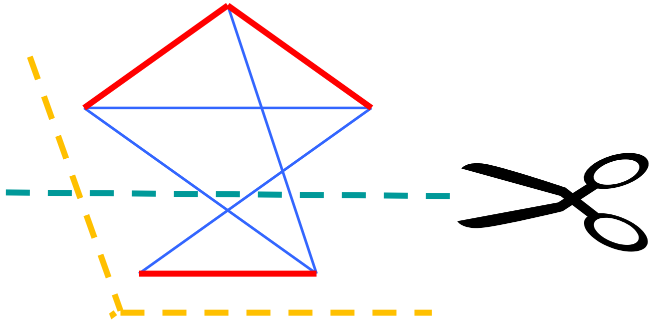
- **Lemma**: Each component is connected with edges with nonnegative weights (Intuitively clear).

If otherwise …

Separating the part

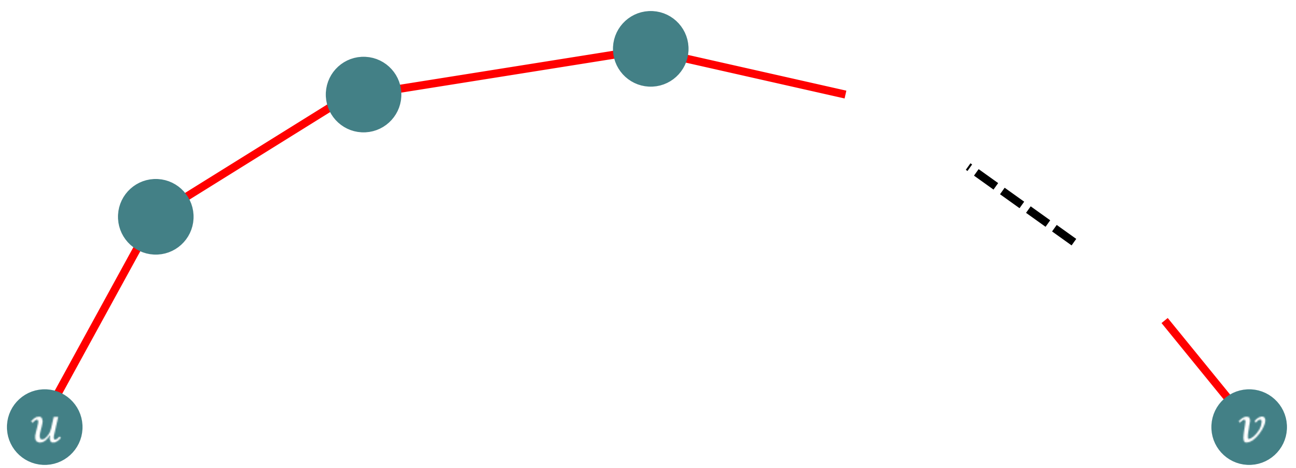
$\Rightarrow$ feasible and better solution.

$\Rightarrow$ contradiction!



Nonnegative weighted path

- By the trivial lemma, for each component V_i and each pair $u, v \in V_i$, $\exists$ path from u to v on $\{\, e : x^*(e) = 1,\ c(e) \geq 0\}$



Assumption

- Now, suppose that M includes all the constraints which do **NOT** satisfy **(RC)**.

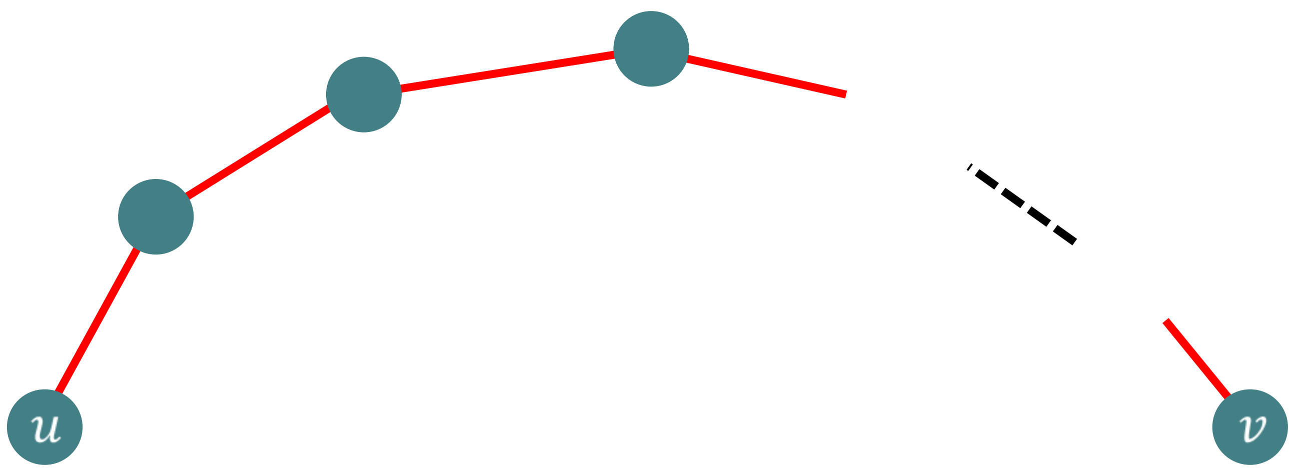
Recall: Given $\underline{x(e)} + \underline{x(f)} - x(g) \leq 1$, **(RC)** requires that $c(e) < 0$ **and** $c(f) < 0$.

- In other words, now, $\underline{x(e)} + \underline{x(f)} - x(g) \leq 1$ is included in M if $c(e) \geq 0$ **or** $c(f) \geq 0$.

Applying the transitivity constraints

- $c(e) \geq 0$ **or** $c(f) \geq 0 \Rightarrow \exists x(e) + x(f) - x(g) \leq 1 \in M.$

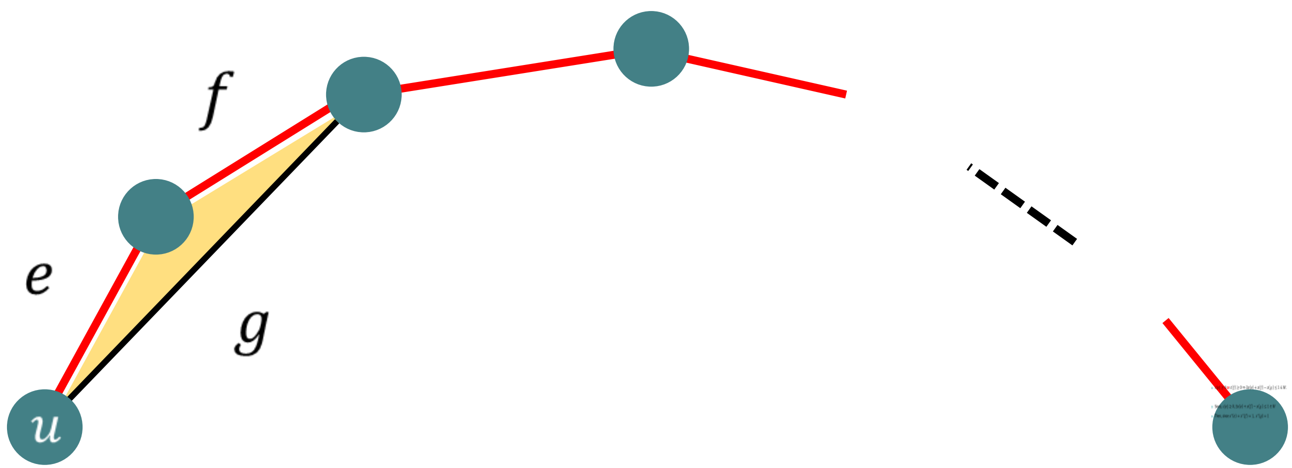
- With this assumption, we have $x^*(u, v) = 1$.
- Let's confirm this.



Applying the transitivity constraints

- $c(e) \geq 0$ **or** $c(f) \geq 0 \Rightarrow \exists x(e) + x(f) - x(g) \leq 1 \in M$.

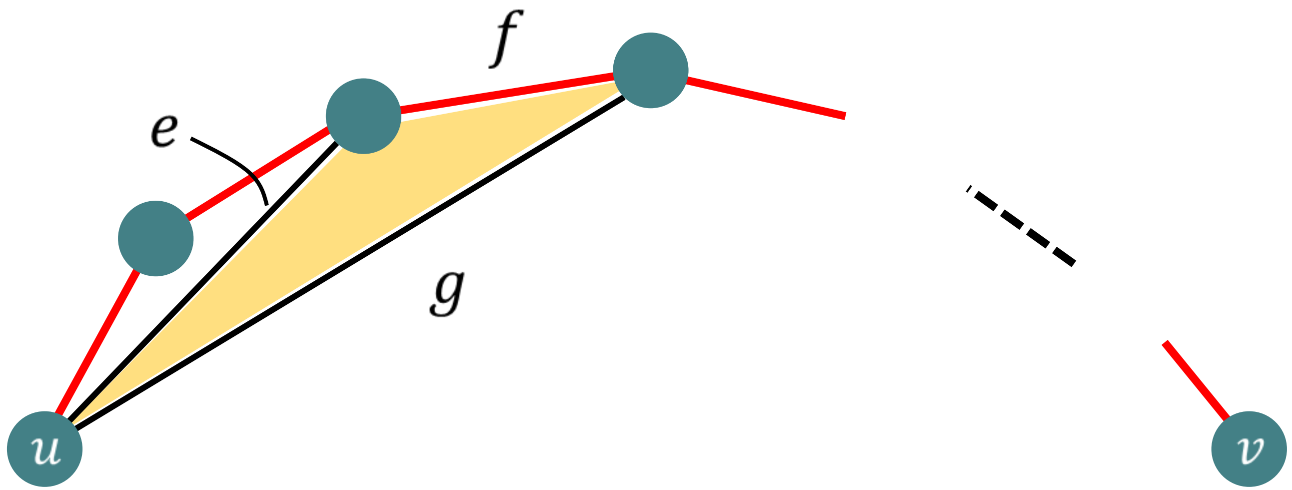
- Since, $c(e) \geq 0$, $\exists x(e) + x(f) - x(g) \leq 1 \in M$
- Then, since $x^*(e) = x^*(f) = 1$, $x^*(g) = 1$



Applying the transitivity constraints

- $c(e) \geq 0$ **or** $c(f) \geq 0 \Rightarrow \exists x(e) + x(f) - x(g) \leq 1 \in M$.

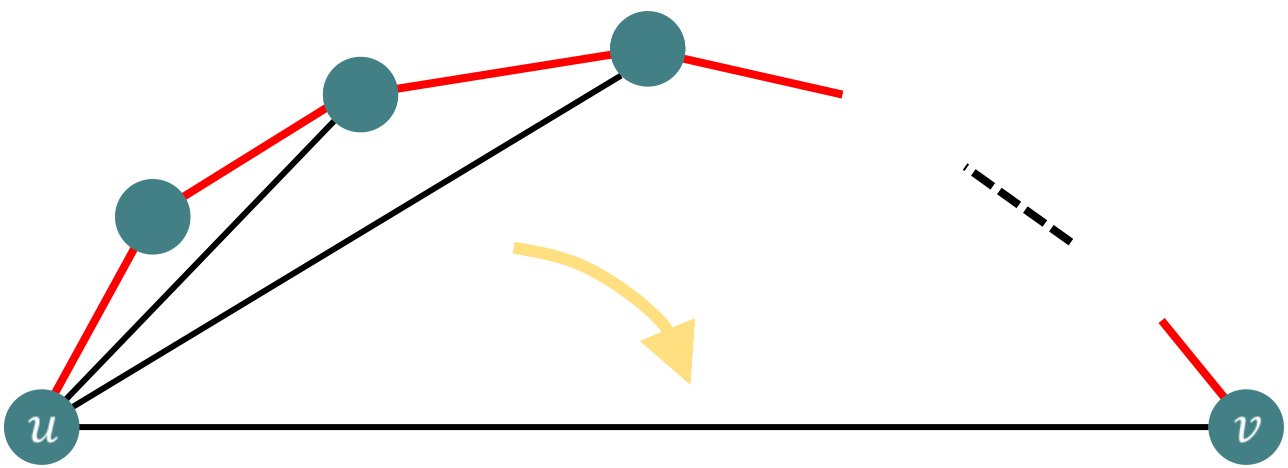
- Since, $c(f) \geq 0$, $\exists x(e) + x(f) - x(g) \leq 1 \in M$.
- Then, since $x^*(e) = x^*(f) = 1$, $x^*(g) = 1$.



Applying the transitivity constraints

- $c(e) \geq 0$ **or** $c(f) \geq 0 \Rightarrow \exists x(e) + x(f) - x(g\ ) \leq 1 \in M.$

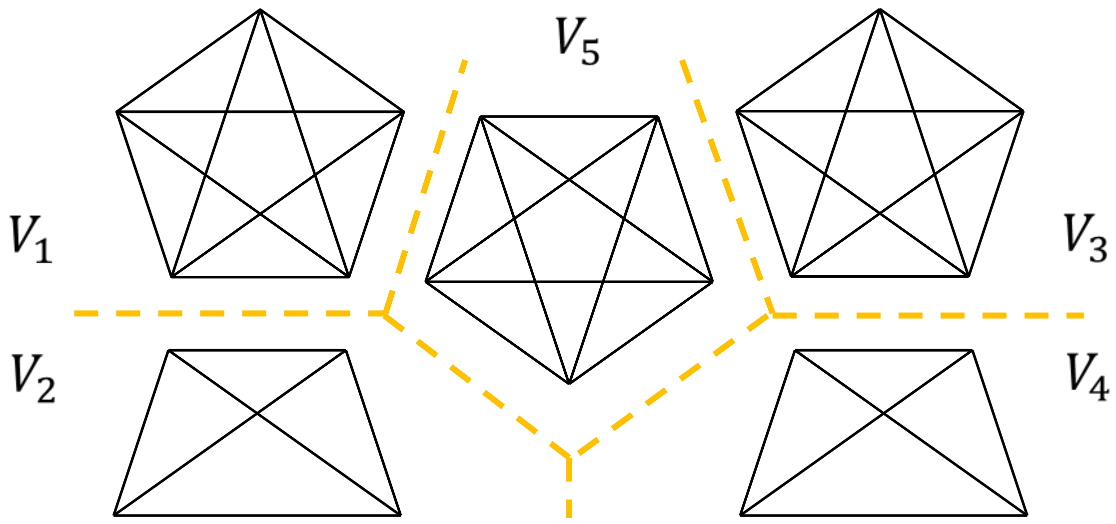
- Finally, we have $x^*(u, v) = 1.$
- This holds for any component and any pair of vertices.



In sum

- This means that each connected component is complete.

  $\Rightarrow x^*$ is feasible to the original CPP.



In sum

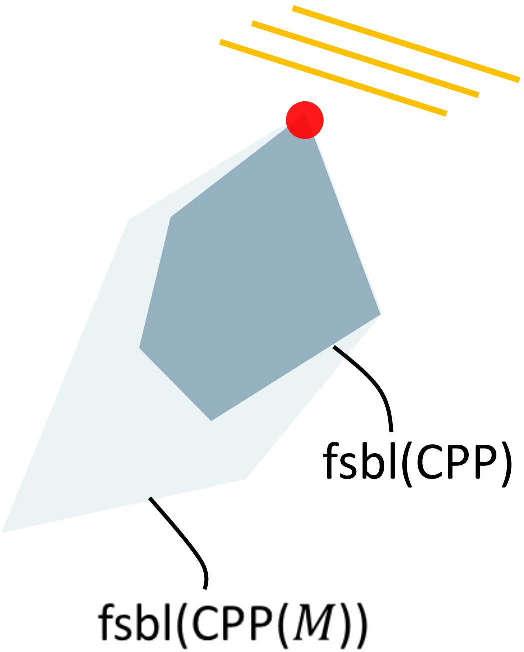
- This means that each connected component is complete.

  $\Rightarrow x^*$ is feasible to the original CPP.

  $\Rightarrow$ opt.sol(CPP(M)) $\subseteq$ opt.sol(CPP)

  $\Rightarrow$ opt.val(CPP(M)) $=$ opt.val(CPP)

  $\Rightarrow$ opt.sol(CPP(M)) $\supseteq$ opt.sol(CPP)



- Thus, with the assumption,

  we have opt.sol(CPP(M)) = opt.sol(CPP).

In other words

- **Recall** (again): the assumption is that all the constraints which do **NOT** satisfy **(RC)** is included in M.

- And we showed that, under this assumption, we have opt.sol(CPP(M)) = opt.sol(CPP).

- Thus, the constraints satisfying **(RC)** are redundant. (That's all for our proof.)

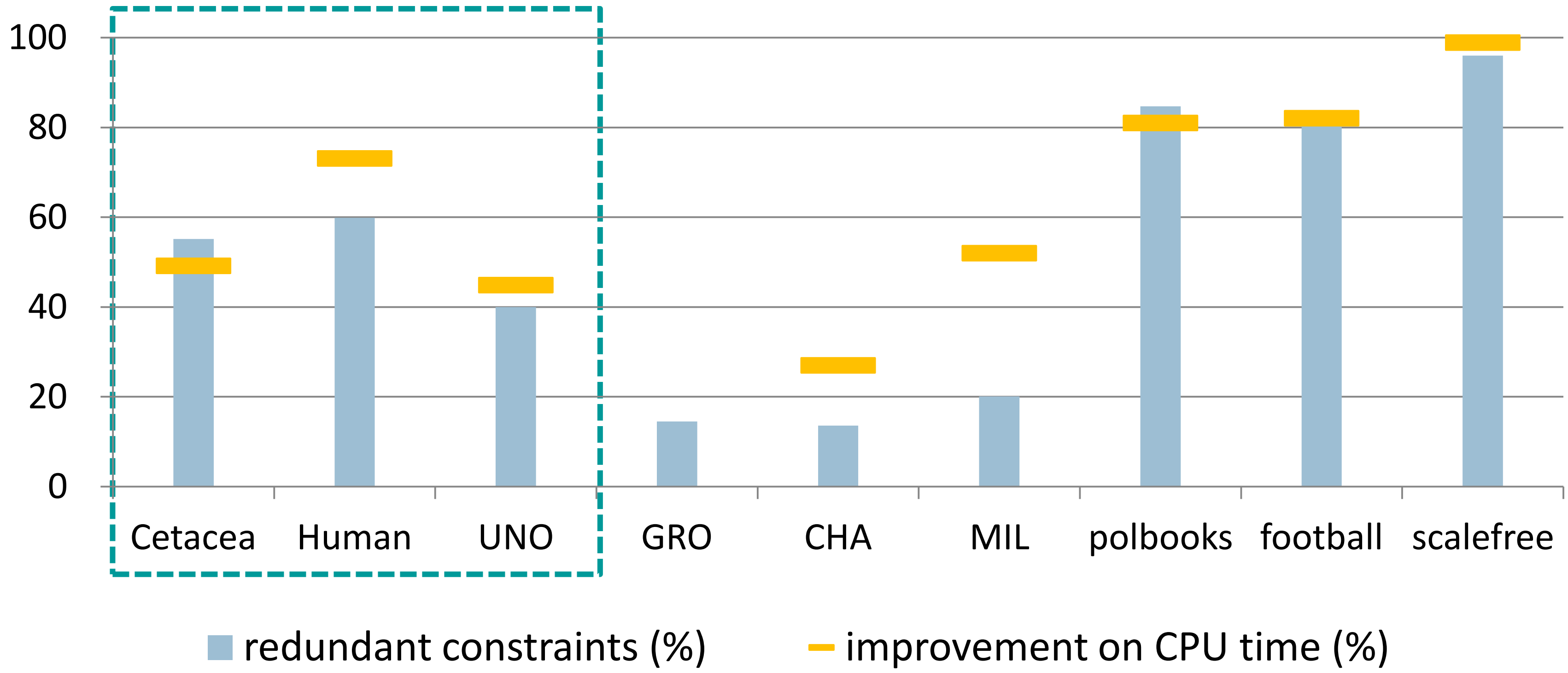
Numerical experiments

- The objective of these numerical experiments is to answer the following two questions.
  - In real-world instances, how many constraints satisfy the condition **(RC)** ?
  - How the computation time will change if we delete the redundant constraints in the IP formulation?

- To solve IP, we use the Gurobi Optimizer 4.5.0.

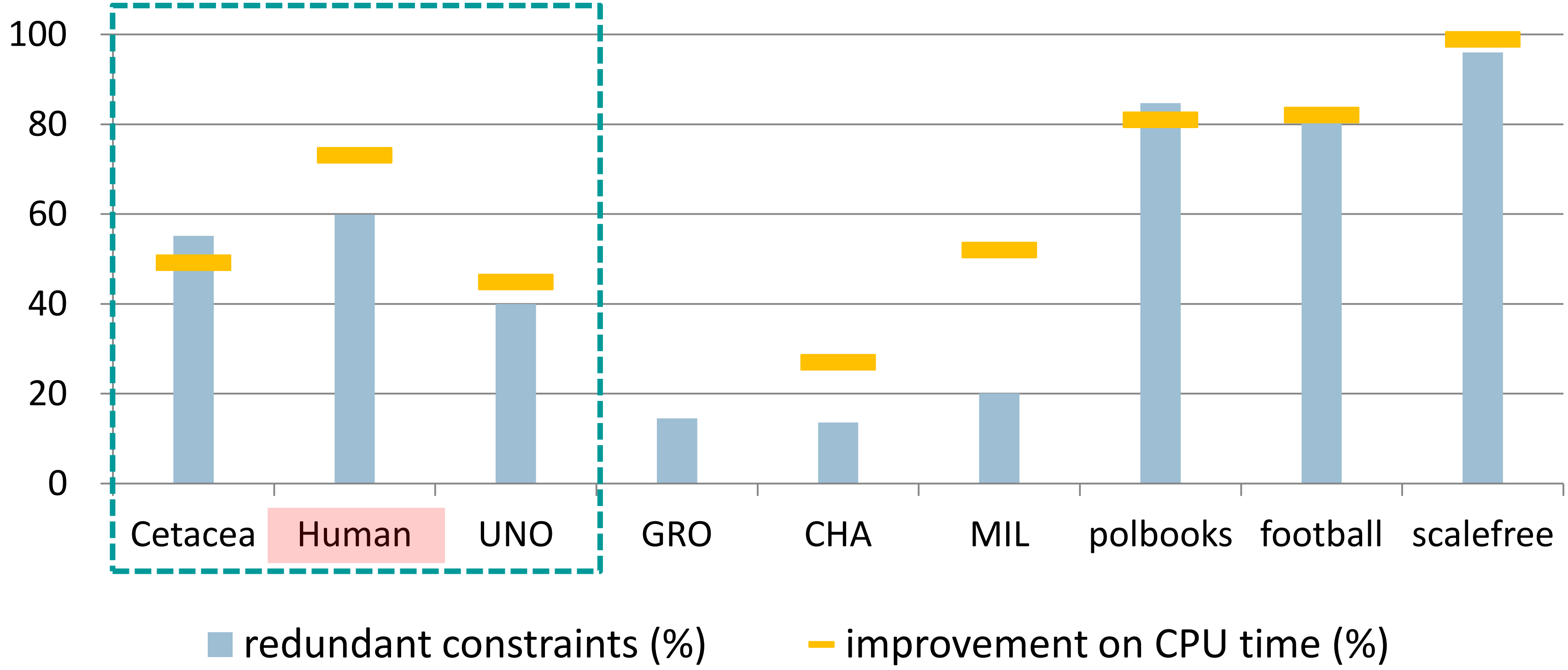
Instances in the Aggregation problem

- The origin of CPP lies in this problem [GrWa ’89].
- One of the important applications of this problem is for qualitative data analysis [BrKö ’06].



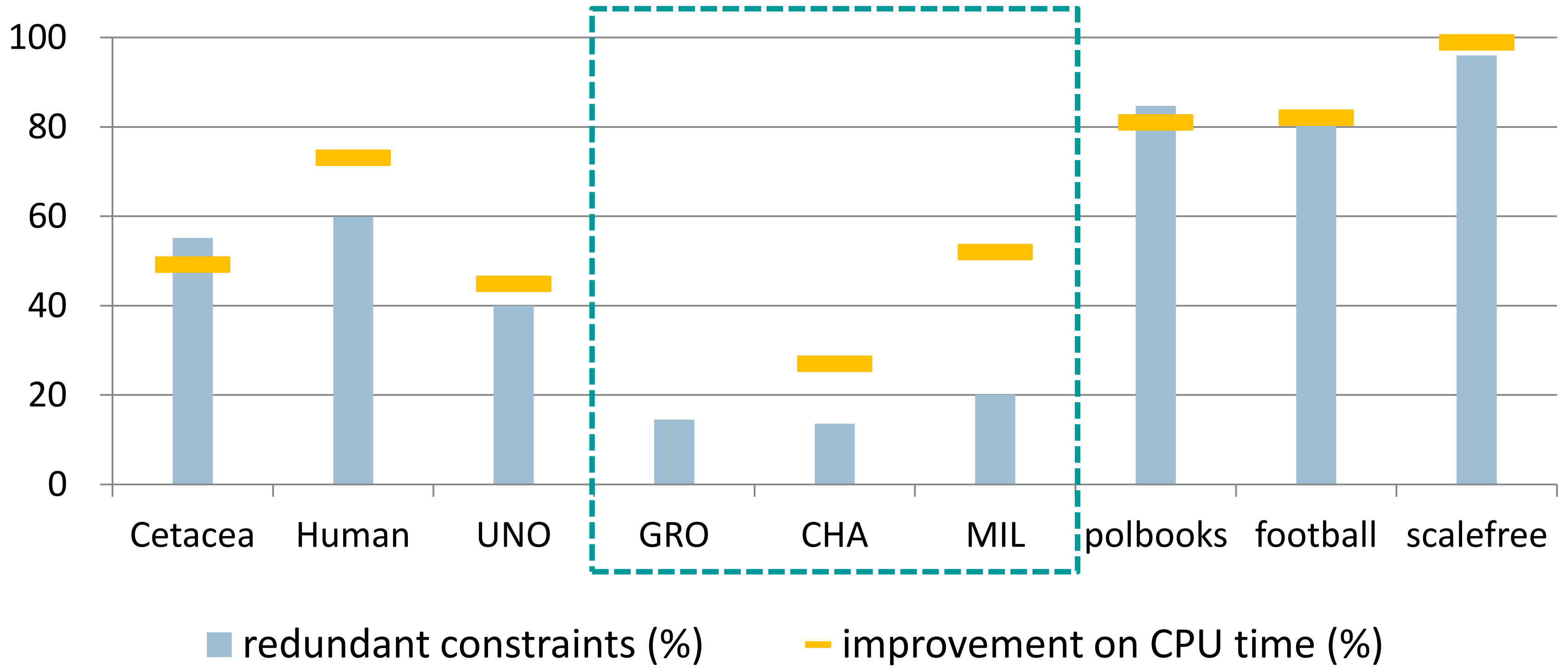
Instances in the Aggregation problem

- Look at the instance ``Human'' ($n = 136$)
  - 60% of the constraints are redundant.
  - Deleting these lessens the computational time to 70%!



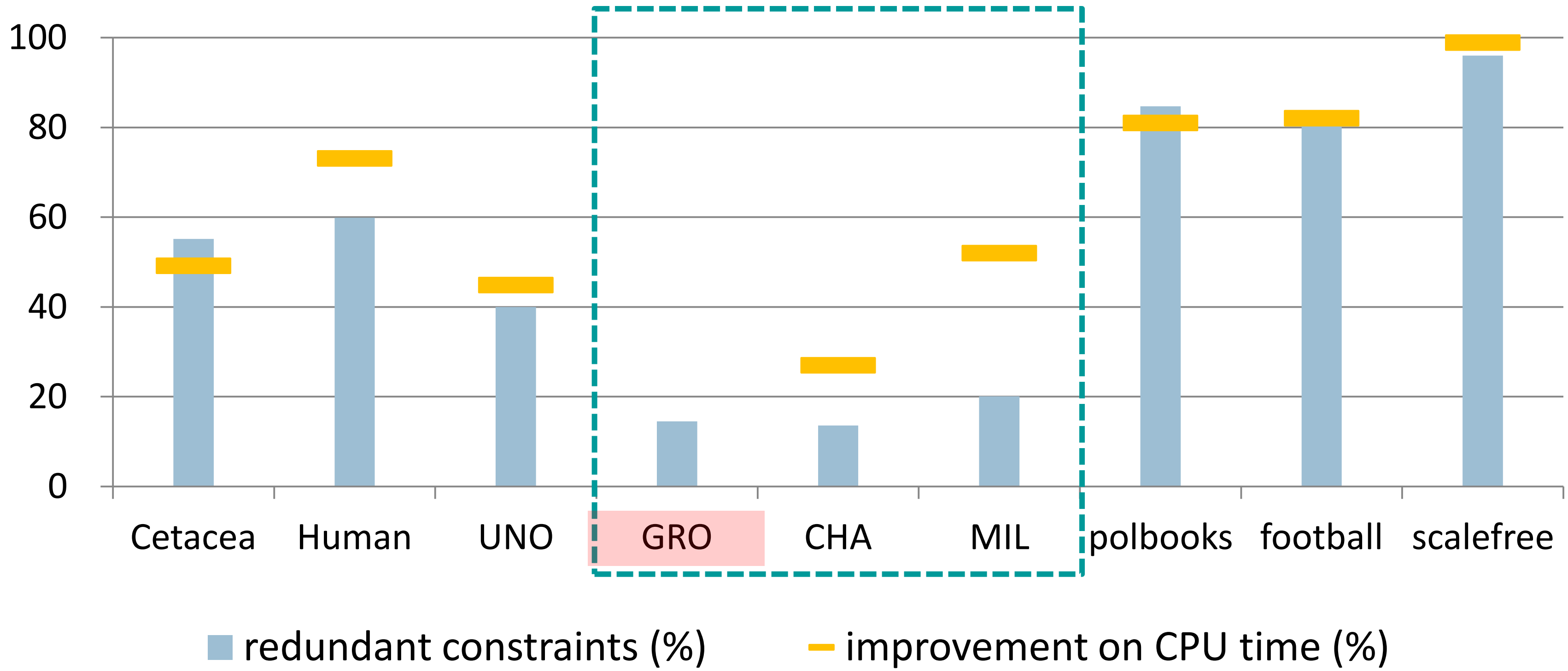
Instances in the Group technology problem

- This problem arises in the manufacturing systems [OoRuSp ’01].
- Compared to the previous ones, a little bit harder.



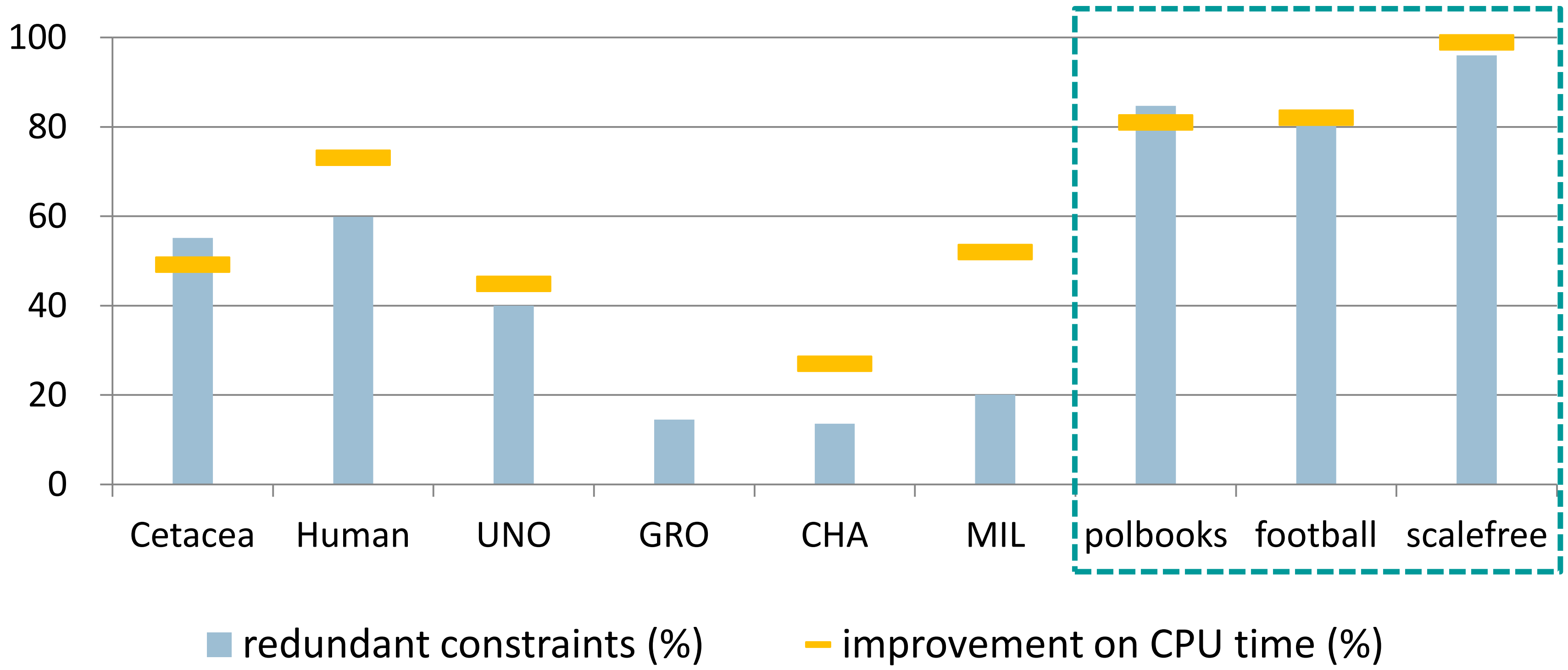
Instances in the Group technology problem

- Look at the instance ``GRO'' ($n = 43$)
  - Deleting the redundant constraints has increased the computation time (More specifically, about 3 times!).



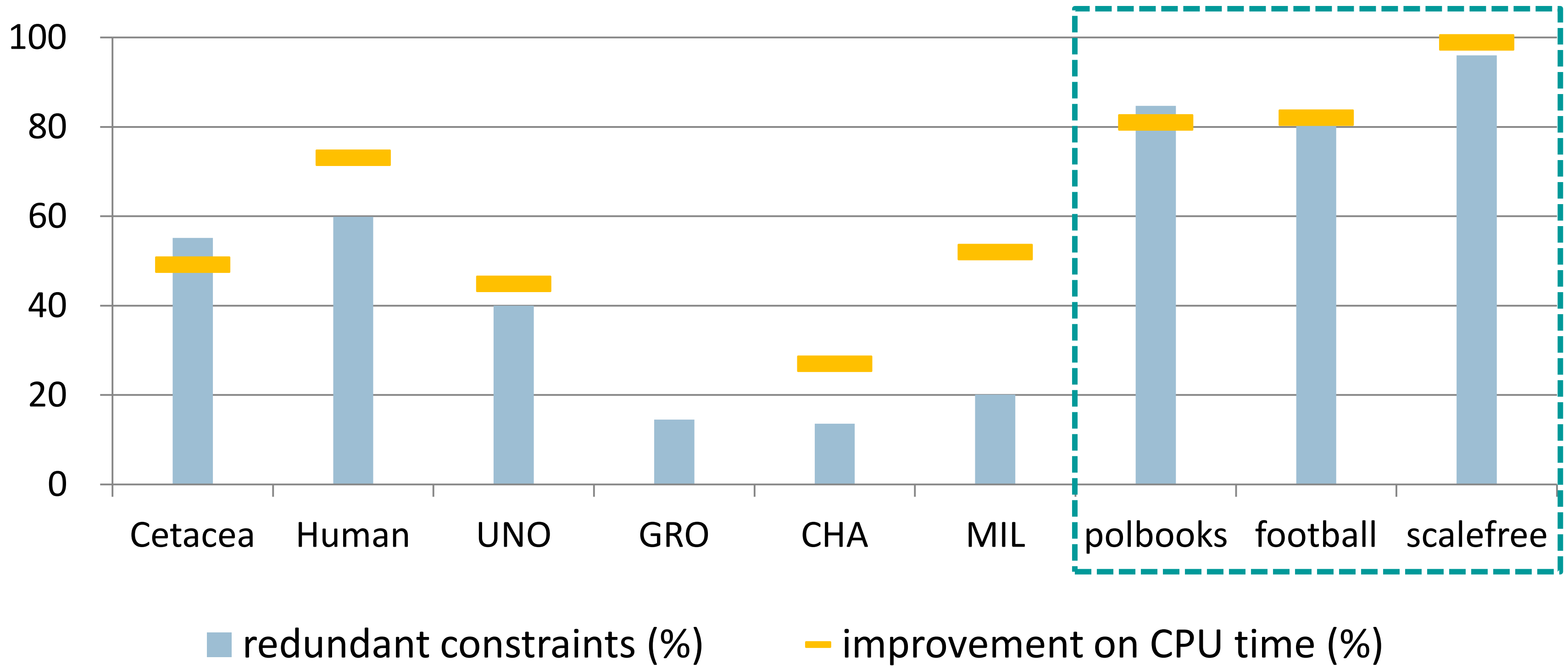
Instances in the Modularity maximization problem

- The most well-studied special case of CPP.
- This problem has attracted very much attention especially for community detection.



Instances in the Modularity maximization problem

- As also reported in [DiTh ’11], the performance is pretty good.
- Though our redundant constraints includes those revealed in [DiTh ’11], the difference is very little.



Final remarks

- We show a special class of redundant constraints in the standard formulation to CPP.
  - These redundant constraints can be found efficiently.
  - Though not mentioned in my presentation, the redundant constraints are also redundant to the LP-relaxation. (Note that, in general IP, this is not true.)
- we are now interested in the limitation of this approach.

Question

- The transitivity constraints $x(e)+x(f)-x(g)\le 1$ satisfying

$c(e) < 0$ **and** $c(f) < 0$

Redundant

$c(e)$ **+** $c(f) < 0$

We are not sure…

$c(e) < 0$ **or** $c(f) < 0$

NOT Redundant

Question

- The transcritivity constraints $x(e)+x(f)-x(g)\le 1$ satisfying

Probably the numerical experiment is inappropriate.

Prof. K

Redundant

$c(e) + c(f) < 0$

The assertion may be true.

$c(e) < 0$ or $c(f) < 0$

Prof. I

Thank you!!!