On the Dynamic Approximate Multicommodity Flow Problem

Radu Cârpa
radu.carpa@ens-lyon.fr

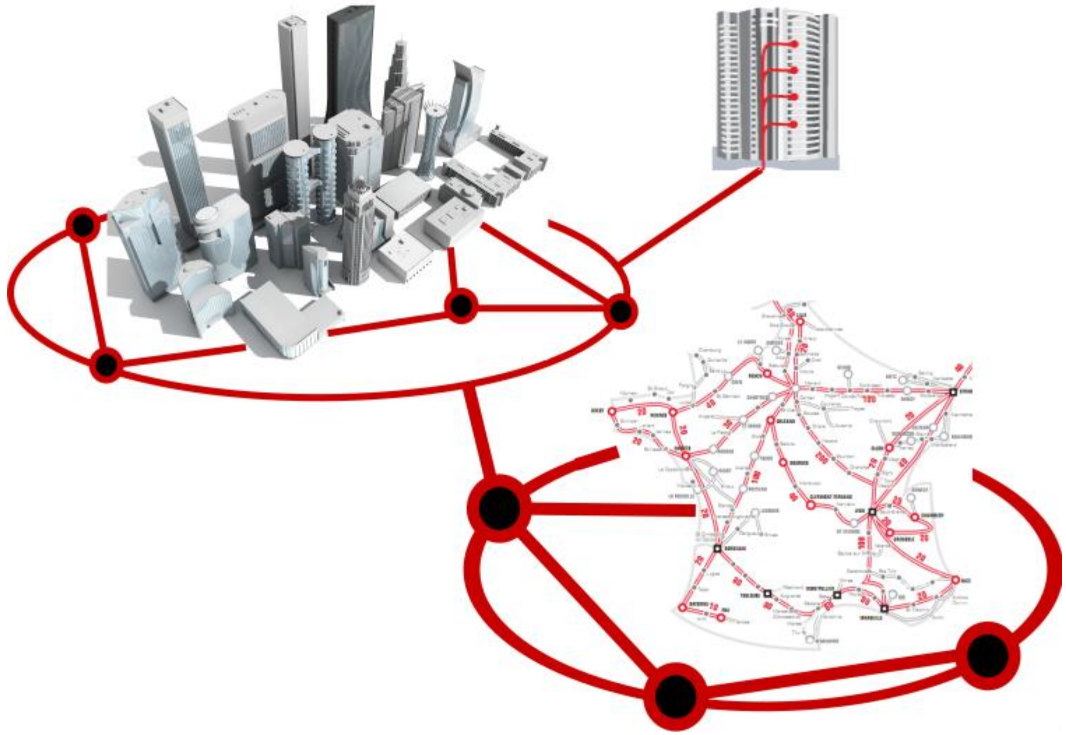
INRIA AVALON / LIP
Ecole Normale
Supérieure de Lyon

CHIST-ERA STAR

0

Context

Energy consumption of computer networks



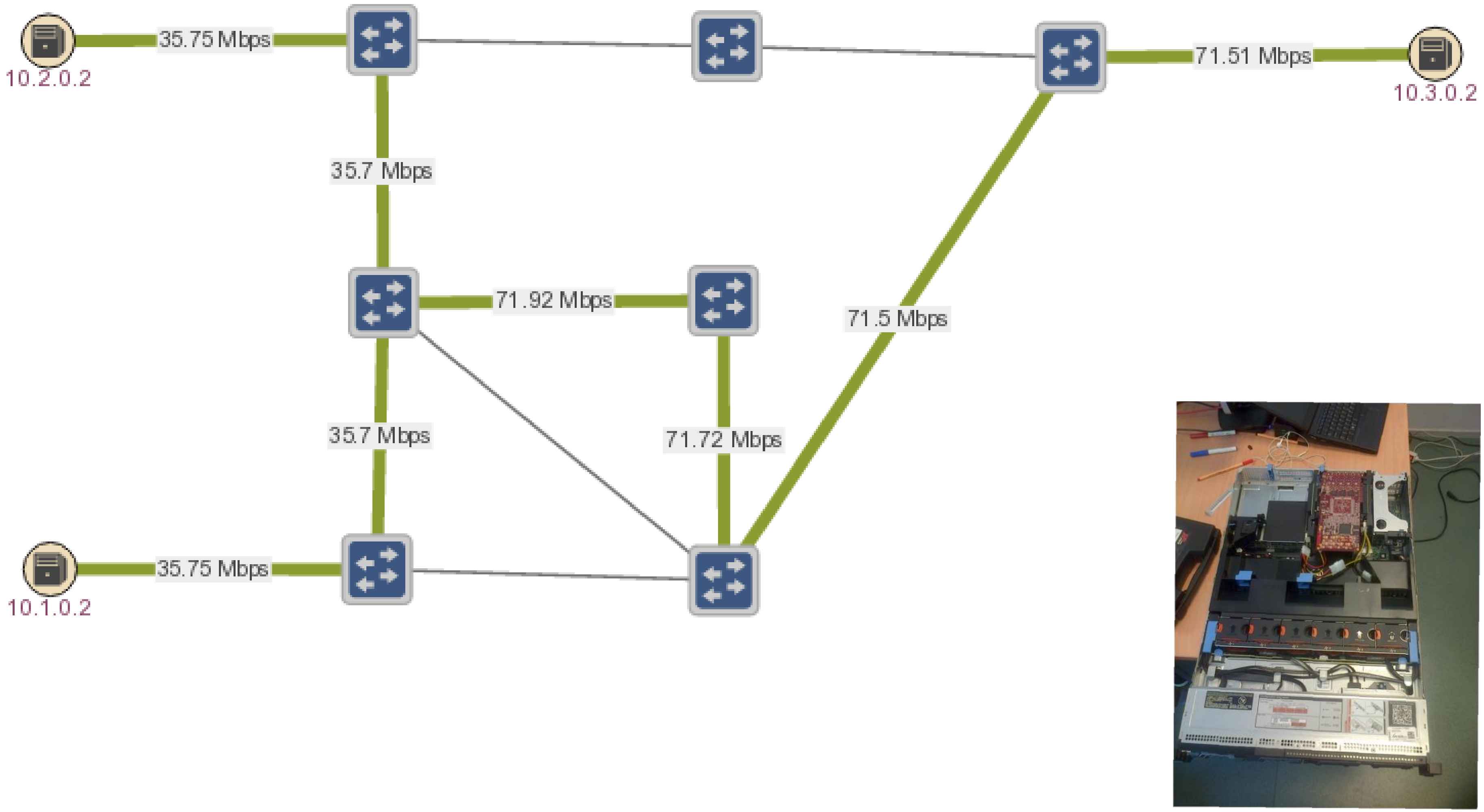
gwatt.net, 2013

Turn-off to save energy

	Power consumption [Watt]
1 Gbps port	7 W
2.5 Gbps port	15 W
10 Gbps port	34 W
40 Gbps port	160 W
100 Gbps port	360 W
400 Gbps port	(1236 W)
1 Tbps port	(2794 W)

*Van Heddeghem, Ward, Filip Idzikowski et al.. 2012. "Power Consumption Modeling in Optical Multilayer Networks." Photonic Network Communications 24 (2): 86–102

Demo



35.75 Mbps
10.2.0.2
71.51 Mbps
10.3.0.2
35.7 Mbps
71.92 Mbps
71.5 Mbps
35.7 Mbps
71.72 Mbps
35.75 Mbps
10.1.0.2

Outline

1. Single and multi-commodity flow problems
2. Approximations to the maximum concurrent flow problem
    1. Garg-Konemann framework
    2. Optimizations with dynamic graph algorithms
3. Future work

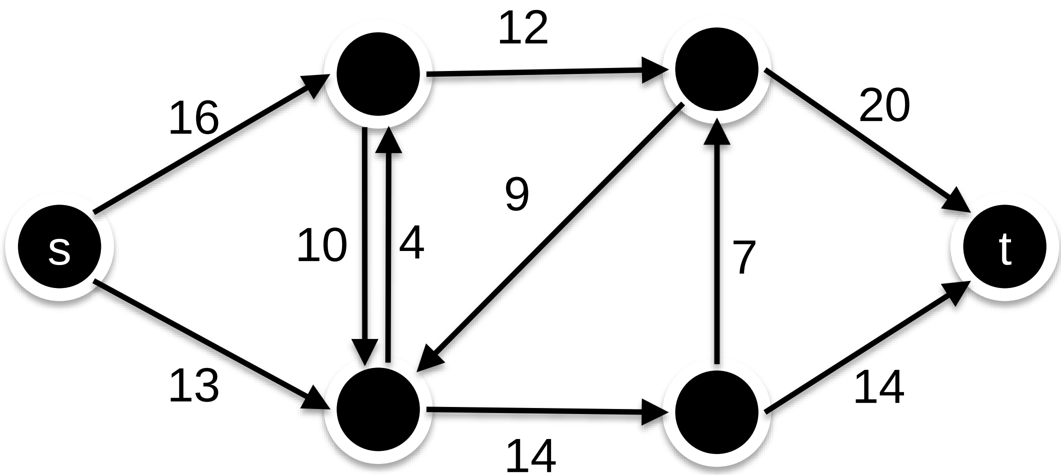
G. Karakostas, Faster approximation schemes for fractional multicom-modity flow problems, ACM Trans. Algorithms 4 (2008) 1V17.

1

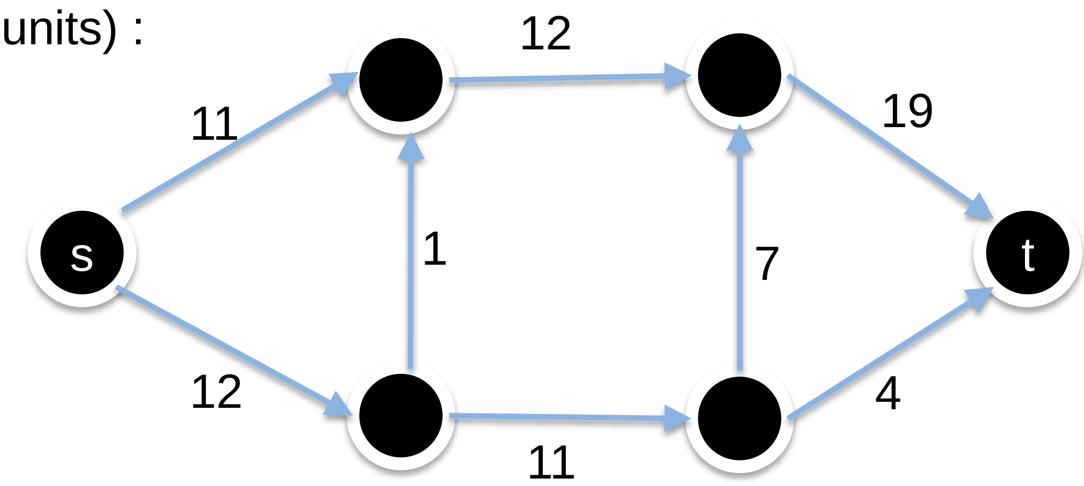
Single and multi-commodity flow problems

Maximum (single-commodity) flow

Capacities $u(e), e \in E$:



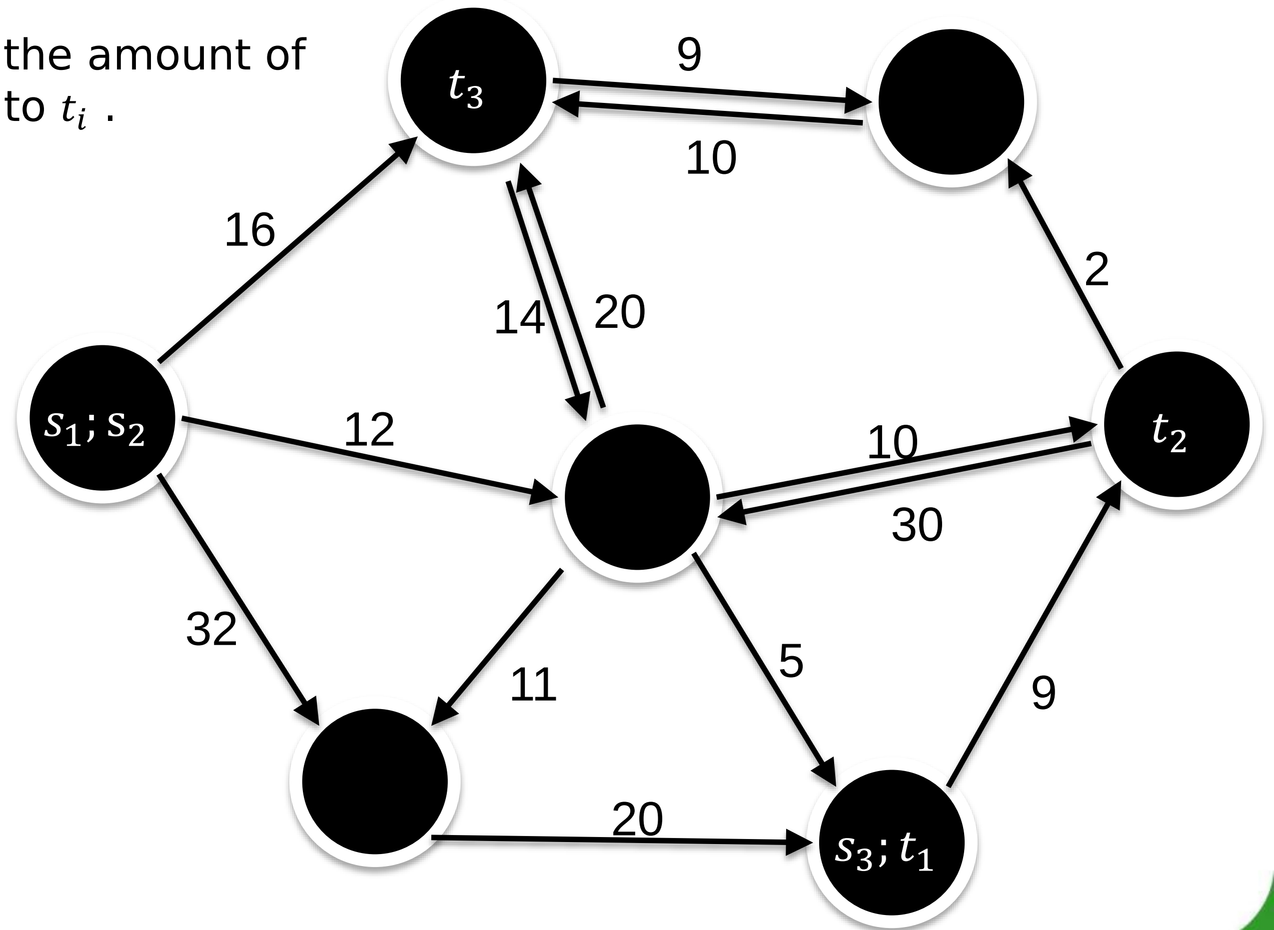
Maximum flow (of 23 total units) :



Maximum multi-commodity flow

k commodities $K_1, \ldots, K_k$ defined by $K_i = (s_i, t_i)$

maximize $\sum_i f_i$, where f_i is the amount of commodity routed from s_i to t_i .

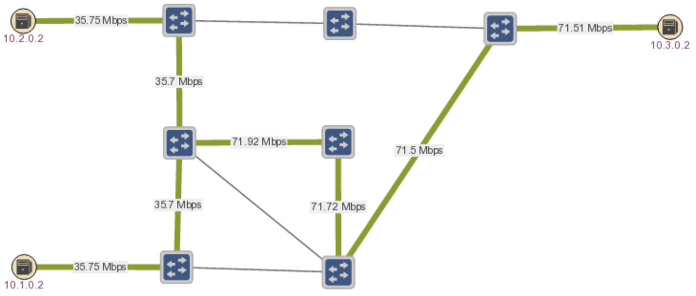


2

Approximations to the maximum concurrent flow problem

Maximum concurrent flow

Multi-commodity flow + k positive demands $\mathrm{d}_1, \ldots, \mathrm{d}_k$



Find the maximum constant λ, such that:
$\forall i$, λd_i units of commodity K_i are routed between s_i and t_i

Garg-Konemann framework:

approximation based on the dual of the linear definition

minimize $D(l) = \sum_e \big(l(e)u(e)\big)$

Garg-Konemann framework

Input: Graph $G = (V, E)$
capacities $u(e)$,
commodity pairs $\{(s_i, t_i)\}_i$ with demands $d_i > 0$,
accuracy parameter $\epsilon > 0$
Output: (Infeasible) flow f

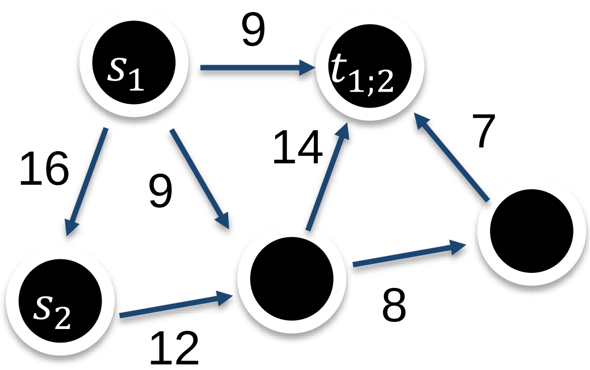
Initialize $f \leftarrow \emptyset$, $l(e) \leftarrow \frac{\gamma}{u(e)}$ for all arcs $e \in E$, where $\gamma = \left(\frac{m}{1-\epsilon}\right)^{-\frac{1}{\epsilon}}$

while $D(l) < 1$ **do**
  for $i := 1, \dots, k$ **do**
    $d_i' \leftarrow d_i$
    while $D(l) < 1$ and $d_i' > 0$ **do**
      Find the shortest path p in P_i
      Find the bottleneck capacity u of p (* $u \leftarrow \min\{d_i', \min_{e \in p} u(e)\}$ *)
      $d_i' \leftarrow d_i' - u$
      Augment the flow f by routing u units of flow along the path p
      foreach arc e in p **do** $l(e) \leftarrow l(e) \cdot \left(1 + \frac{\epsilon \cdot u}{u(e)}\right)$
    end
  end
end

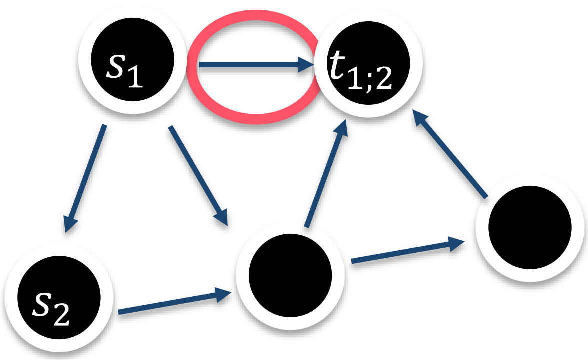
$P_1 = \{\text{all paths from } s_1 \text{ to } t_1\}$

$u \leftarrow d_1' = d_1 = 3$

For $\epsilon = 0.05$, $l(e) = l(e) \cdot 1.016$



$d_1 = 3$
$d_2 = 2$



Garg-Konemann framework
key observations

The shown algorithm finishes after at most $t \coloneqq 1 + \frac{\lambda}{\epsilon}\log_{1+\epsilon}\frac{m}{1-\epsilon}$ phases

The obtained solution must be scaled down by $\log_{1+\epsilon}\frac{1}{\gamma}$

If $\lambda > 1$, the scaled down flow has a value of at least $(1-3\epsilon)\,\lambda$

m: number of arcs

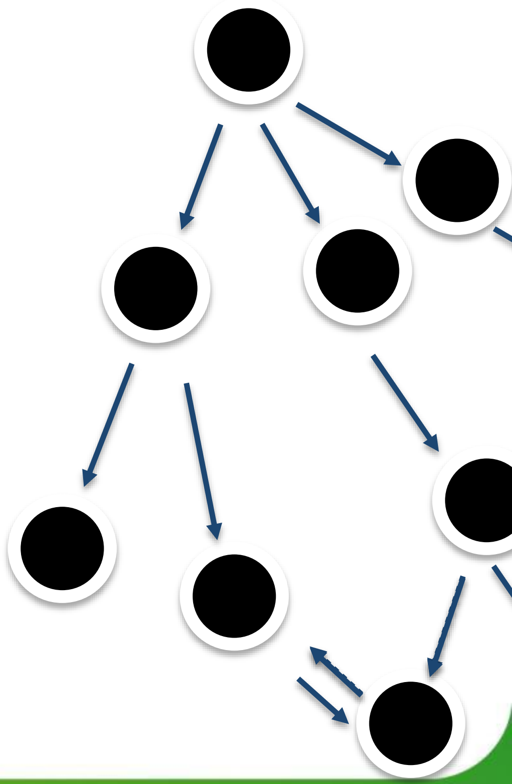
G. Karakostas, Faster approximation schemes for fractional multicom-modity flow problems, ACM Trans. Algorithms 4 (2008) 1V17.

Optimization with dynamic graph algorithms
Solution 1

☹ Shortest path computation takes time

Use dynamic all-pairs shortest paths algorithms :
Partially update the structure when arc weight changes

☹ Still costly updates
☹ Worst case update cost equal to re-computing from scratch



*Camil Demetrescu and Giuseppe F. Italiano. Experimental analysis ofdynamic all pairs shortest path algorithms. ACM Trans. Algorithms,2(4):578–601, October 2006.

Optimization with dynamic graph algorithms
Solution 2

Construct a data structure allowing a fixed O(1) cost per increase of the length of any arc.

based on probabilistic graph sparsification

probabilistic result : probability of n^-5 to have a sup-optimal result

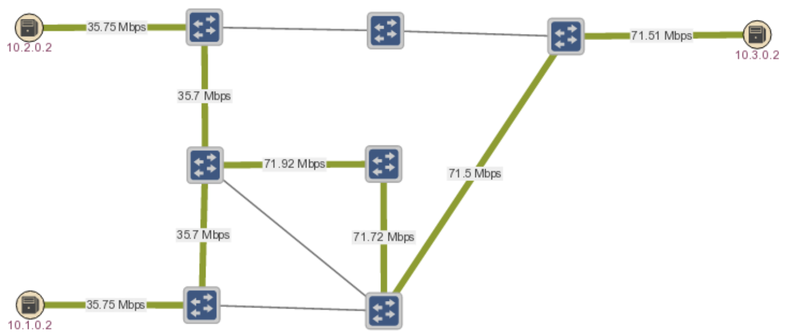
Final complexity : $O((m+k)n\epsilon^{-2}\log M)$

G. Karakostas, Faster approximation schemes for fractional multicom-modity flow problems, ACM Trans. Algorithms 4 (2008) 1V17.

3

Future work

Ideas for future work



To better save energy in computer networks:

Networks are not static, demands permanently change

A fast reaction to network changes is needed

- ☺ Update λ without re-computing the solution from scratch
- ☺ Probabilistic and approximate result is enough if it is fast
- ☺ Computational time limit and getting ϵ as output
- ☺ Fast non-fractional commodity placement ? ☹

Thank you

radu.carpa@ens-lyon.fr