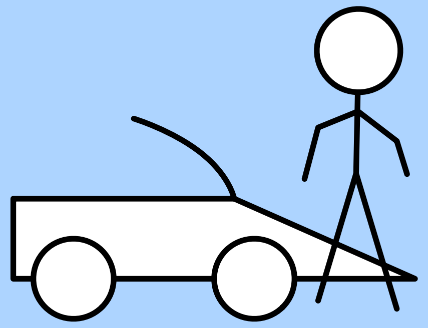
Truthful Mechanisms for Delivery with Agents

Andreas Bärtschi, Daniel Graf, Paolo Penna
Department of Computer Science, ETH Zürich

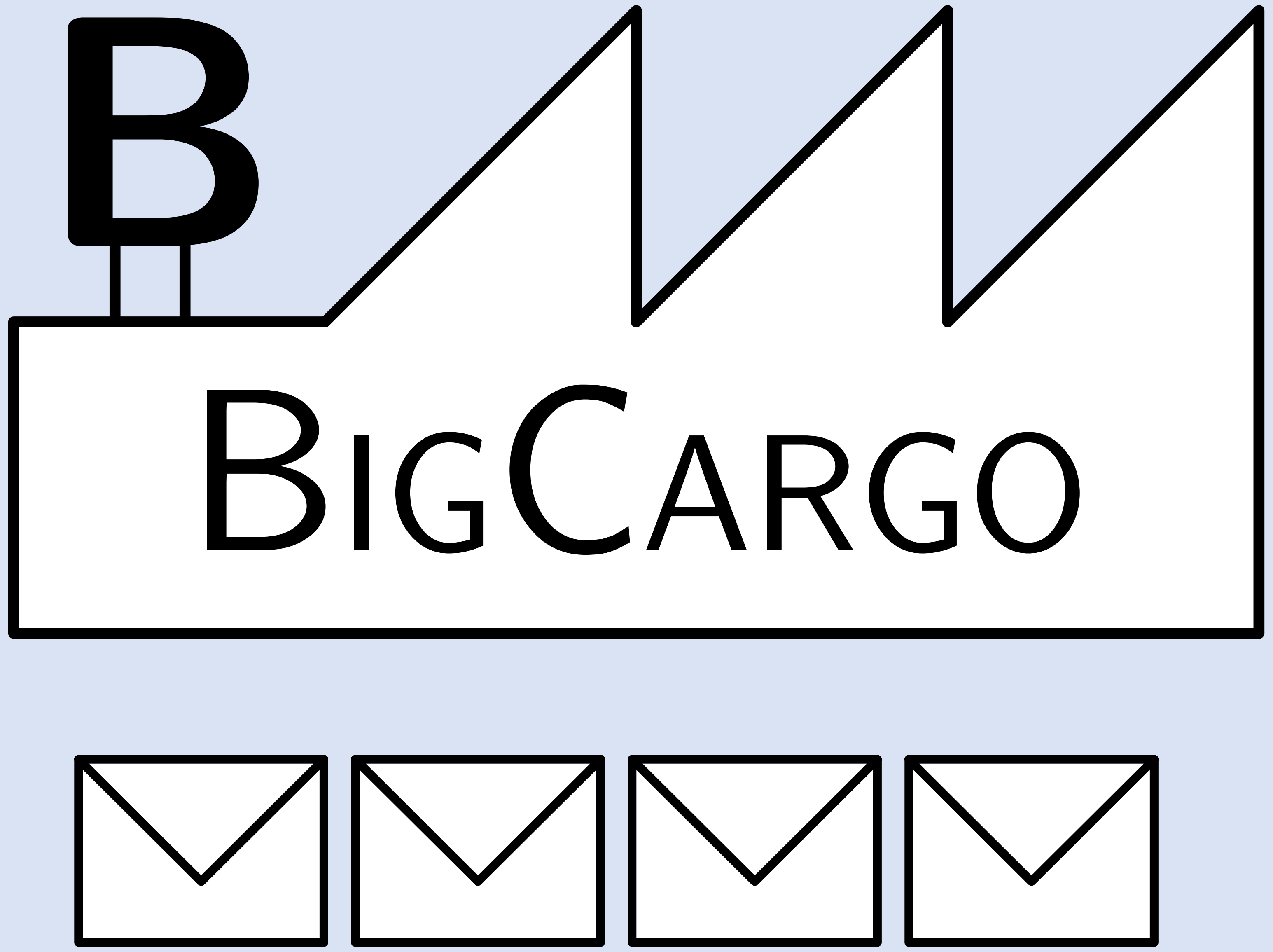
17th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems
September 7-8, 2017 · Vienna, Austria

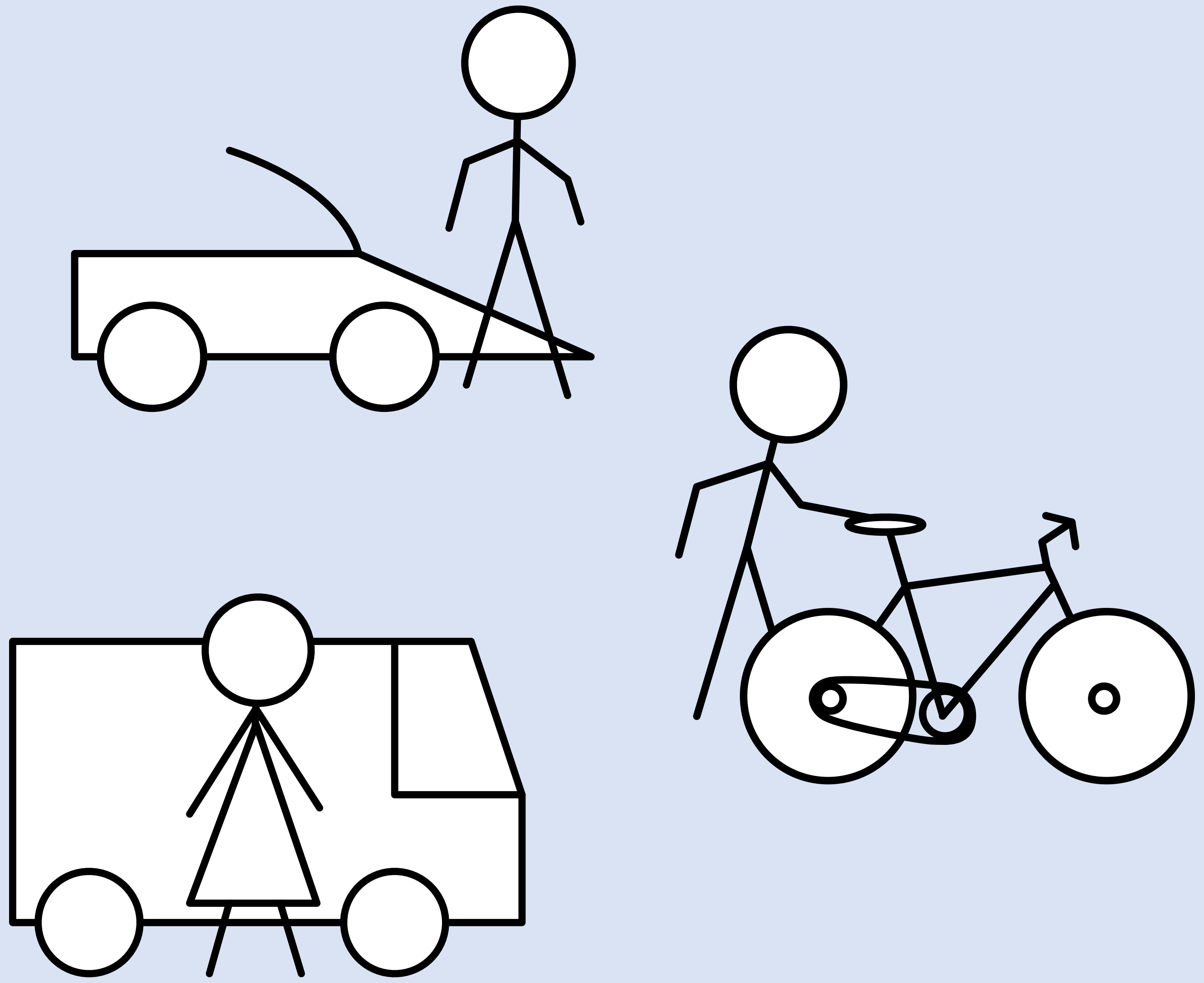
Key Points of this Talk/Paper

- model: cargo company who hires selfish drivers



- turn approximation algorithm into mechanism
- improve the guarantees for certain special cases





packages

pick up & deliver

drivers

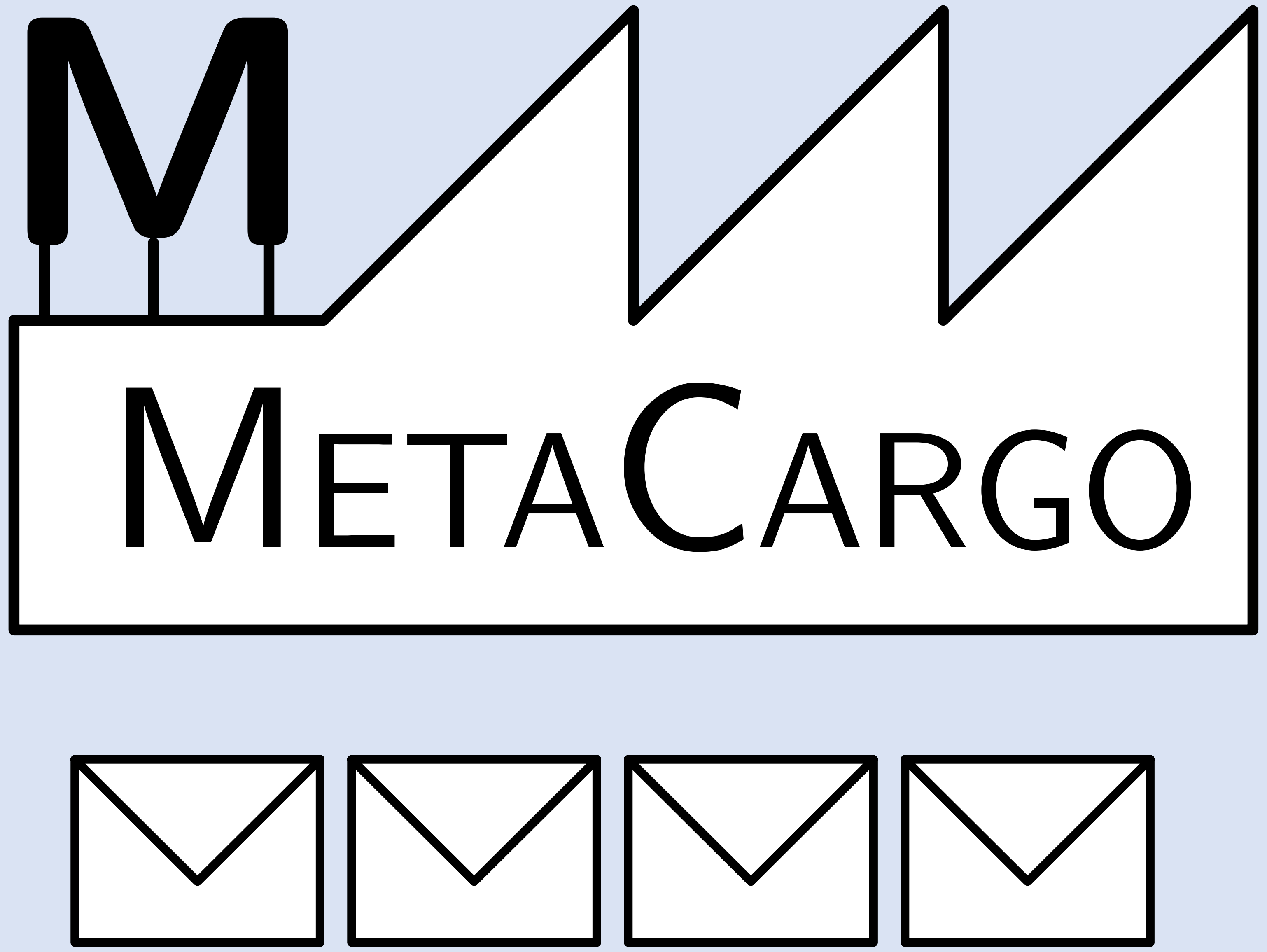
employ & control

Wanted:

energy-optimal delivery schedule

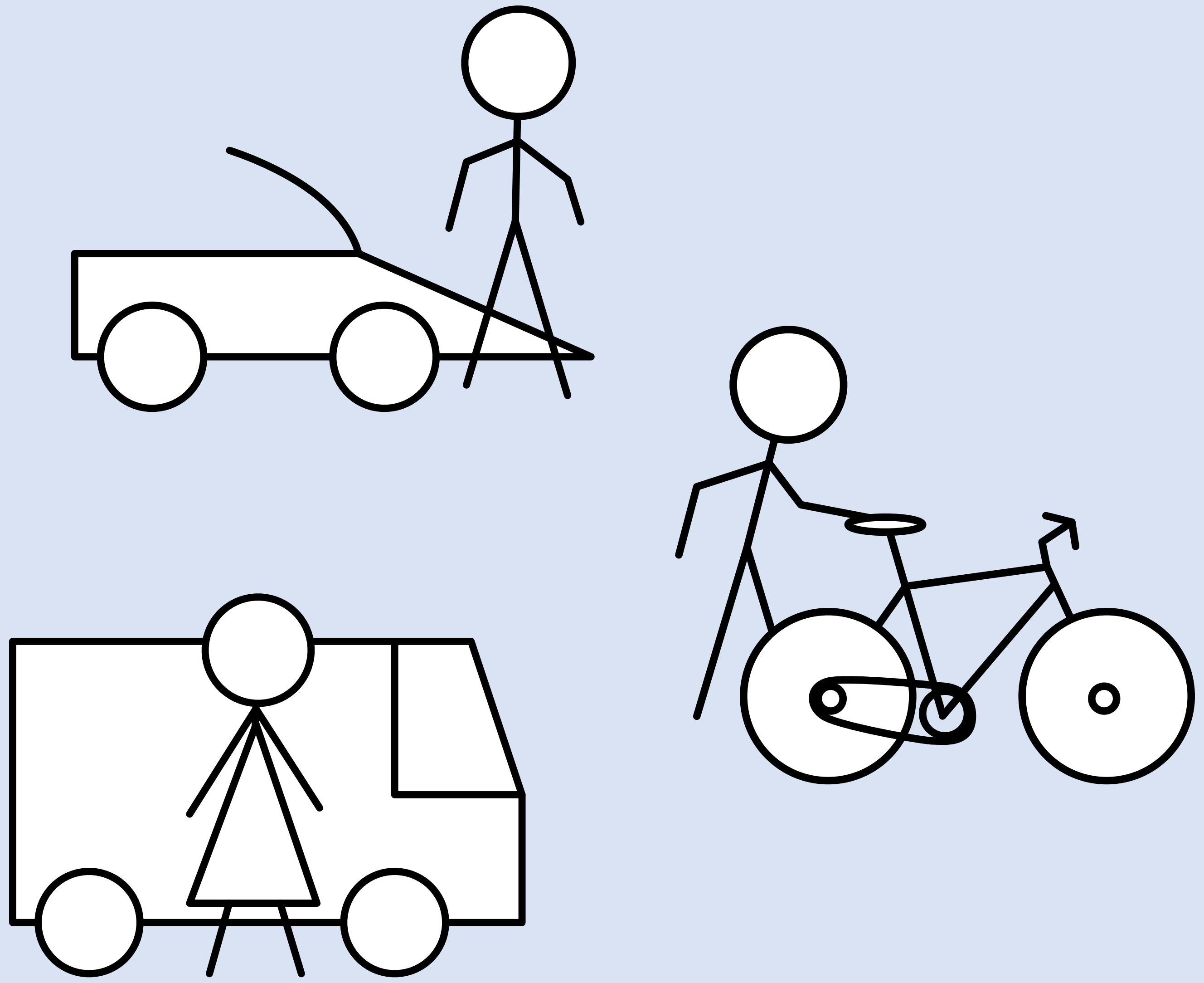
Model by Bärtschi et al. STACS'17

Bärtschi, Chalopin, Das, Disser, Hackfeld, Penna: „Energy-efficient Delivery by Heterogeneous Mobile Agents“. STACS 2017, pages 10:1–10:14, 2017.



packages

pick up & deliver



drivers

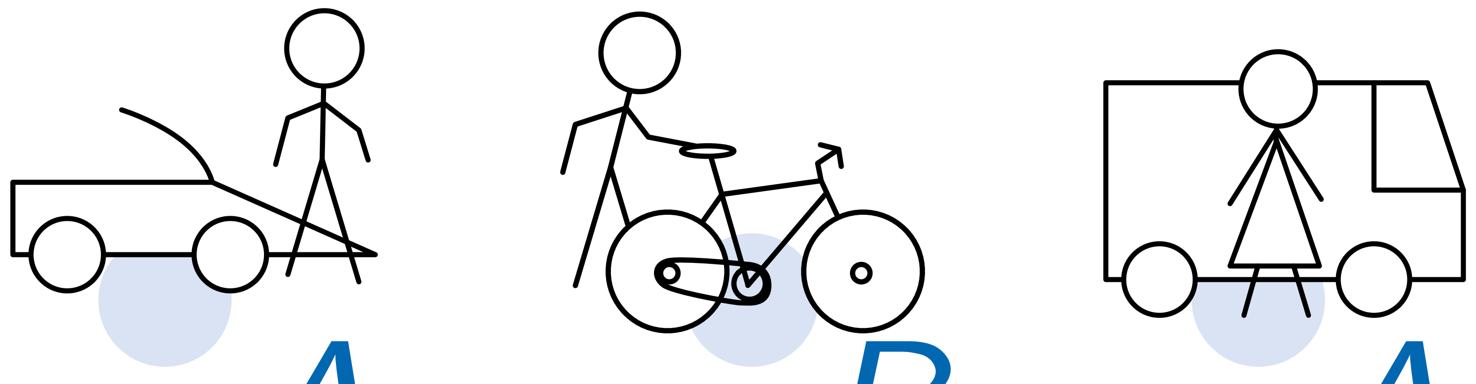
self-employed & selfish

Wanted:

negotiation procedure $=$ decision and pricing mechanism

The Game

Players: METACARGO and individual drivers



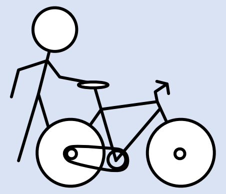
$p_1 = A$ $p_2 = B$ $p_3 = A$

Steps:

1. MetaCargo lists the jobs
2. Drivers announce their costs

$A \rightarrow B$ $C \rightarrow D$

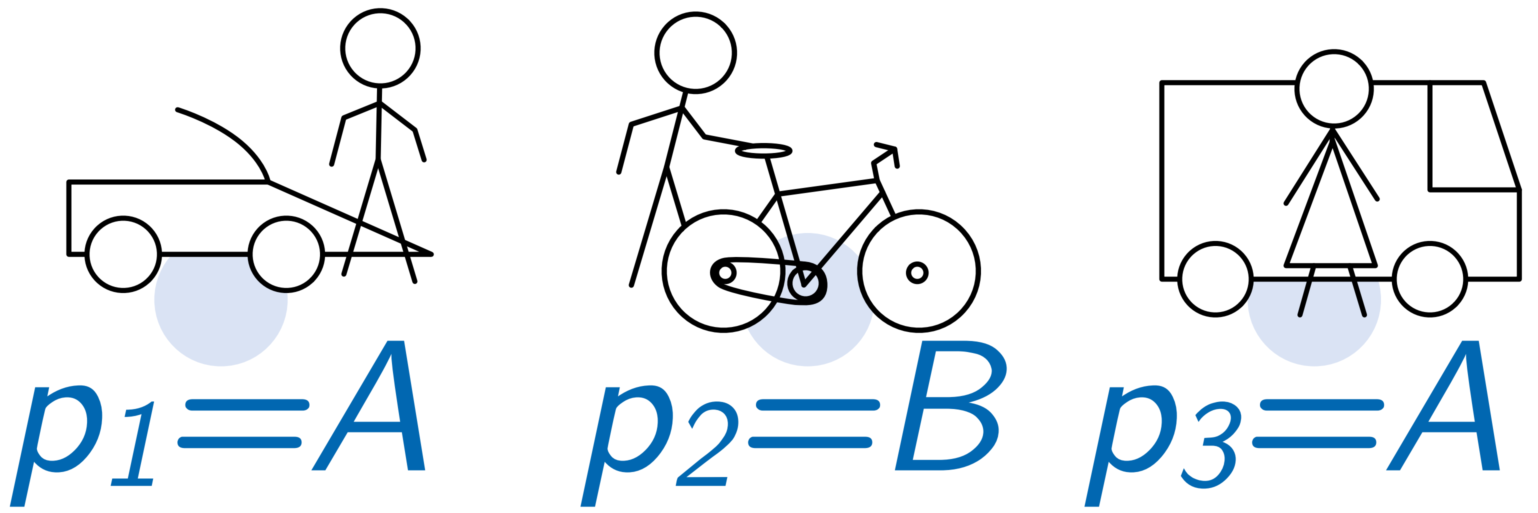
Really?



: „$w_i = 2€/km$“

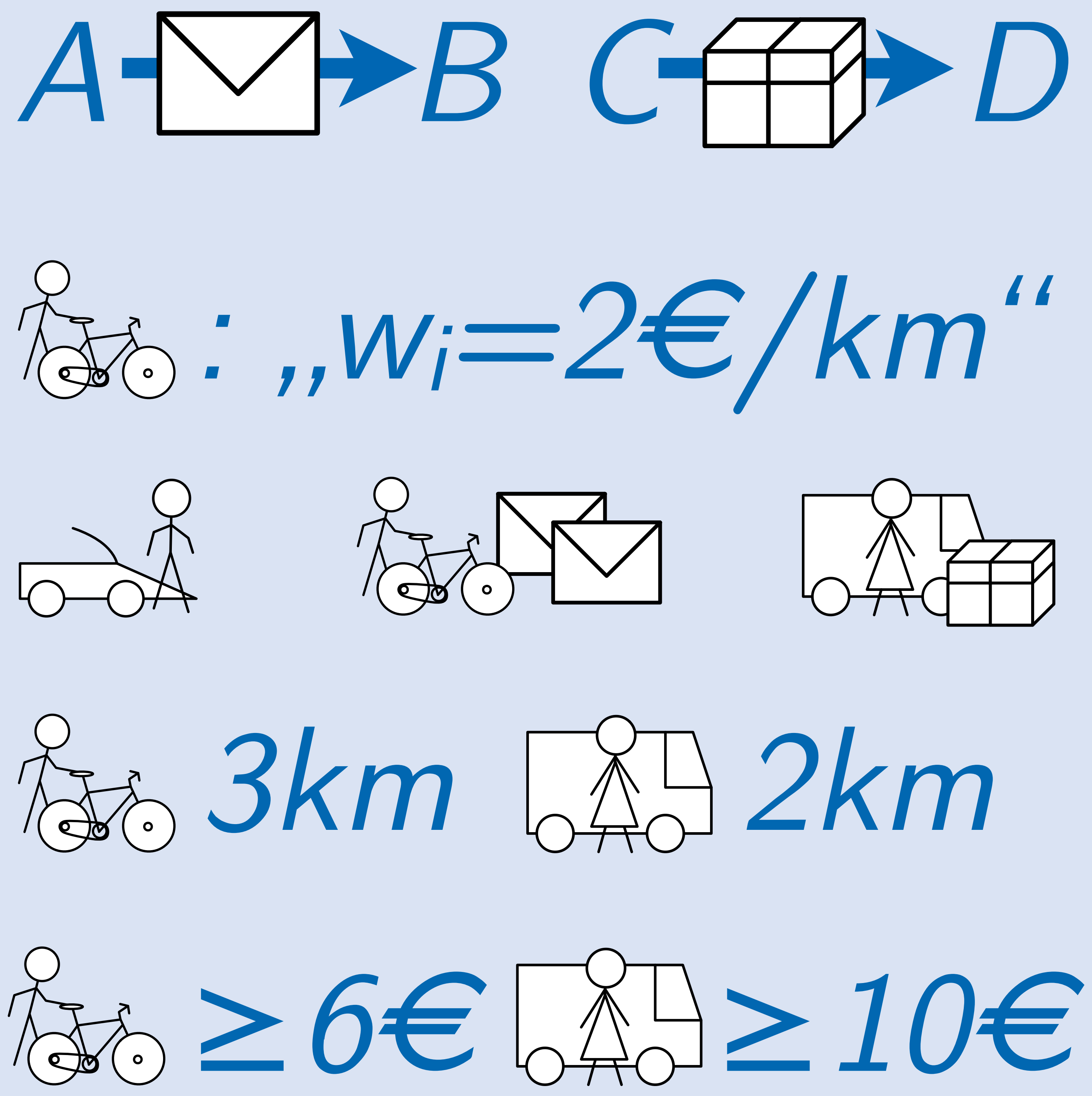
The Game

Players: M METACARGO and individual drivers



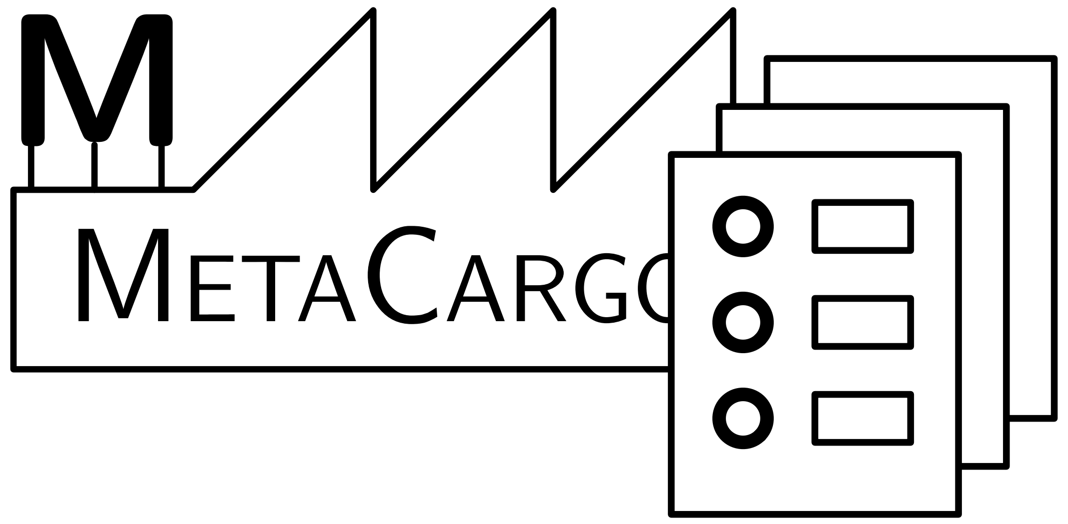
Steps:

1. MetaCargo lists the jobs
2. Drivers announce their costs
3. MetaCargo decides schedule
4. Drivers fulfill their orders
5. MetaCargo pays the drivers

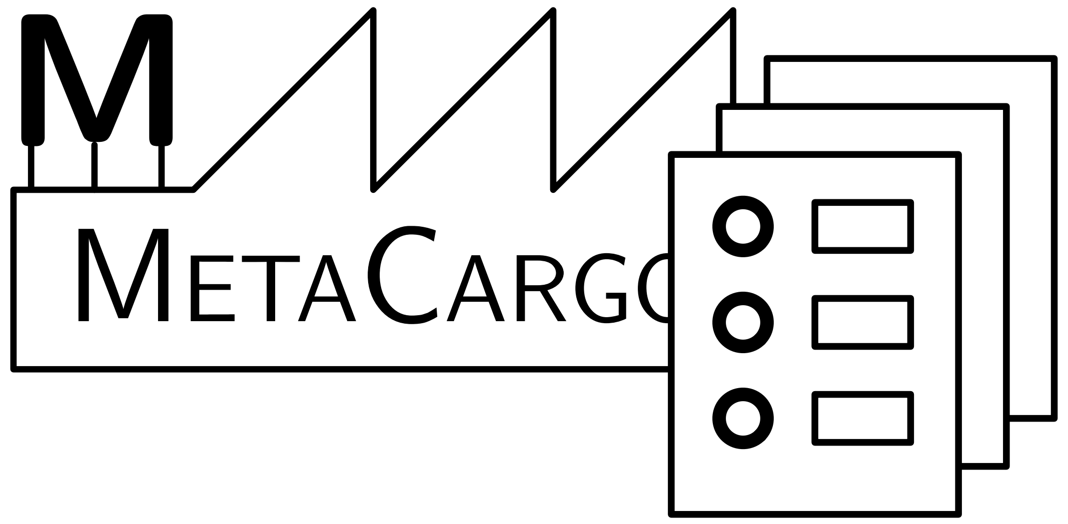


Mechanism for MetaCargo

- publicly known rules, fixed in advance
- fully determines selection and payments



Mechanism for MetaCargo



- publicly known rules, fixed in advance
- fully determines selection and payments

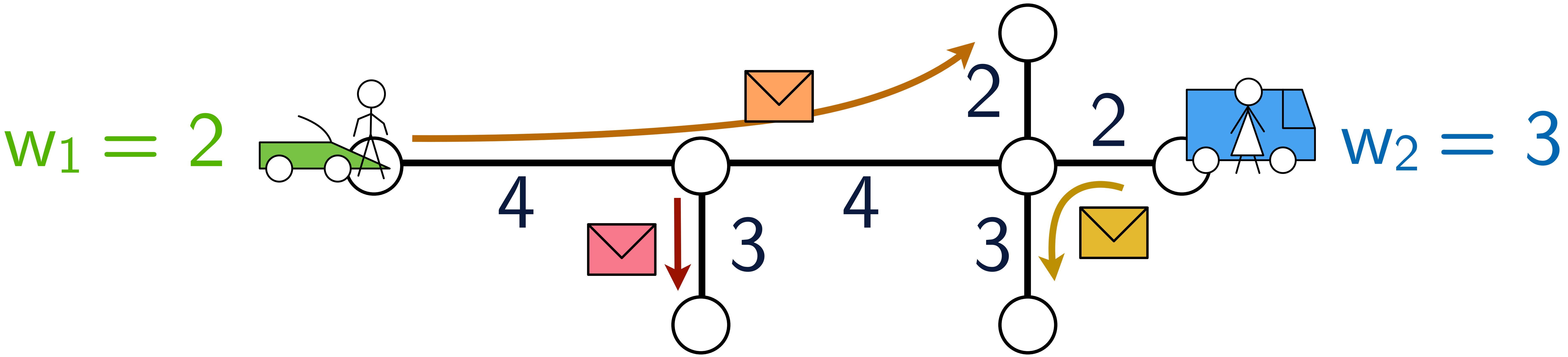
Goals of a good mechanism

1. truthfulness	*lying does not pay off*
2. voluntary participation	*the game is worth playing*
3. near optimality	*costs close to best possible*
4. frugality	*reasonable prices*
5. polynomial running time	*fast to compute*

Setting

- n nodes in graph (each edge with travel distance)
- m packages (each with source and target)
- k agents (each with initial position p_i and weight w_i)
- objective function: $\text{cost}(x) = \sum_{i=1}^{k} w_i \cdot d_i(x)$

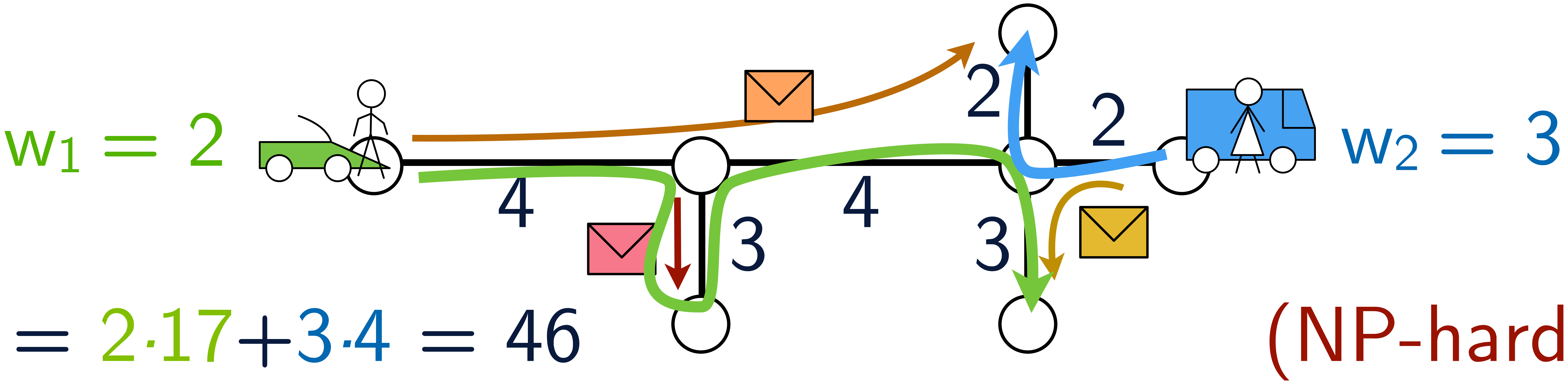
Example



Setting

- n nodes in graph (each edge with travel distance)
- m packages (each with source and target)
- k agents (each with initial position p_i and weight w_i)
- objective function: $\text{cost}(x) = \sum_{i=1}^{k} w_i \cdot d_i(x)$

Example

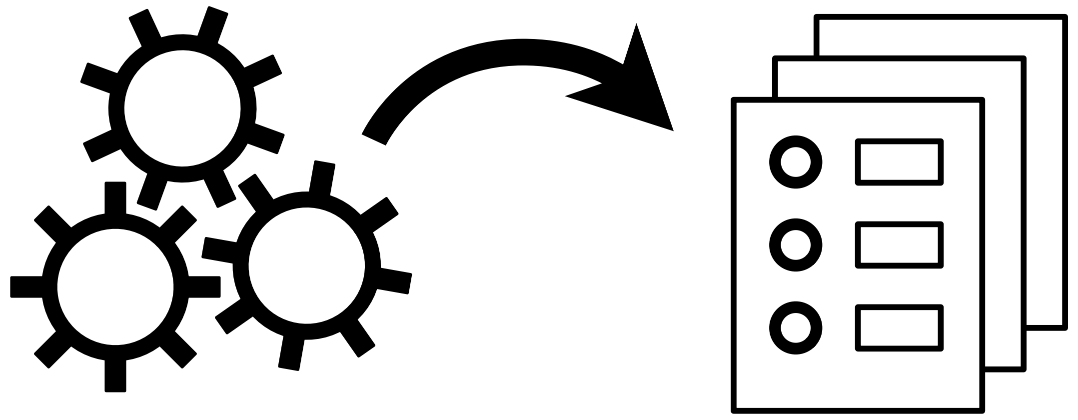


$\text{cost}(\text{OPT}) = 2{\cdot}17 + 3{\cdot}4 = 46$

(NP-hard)

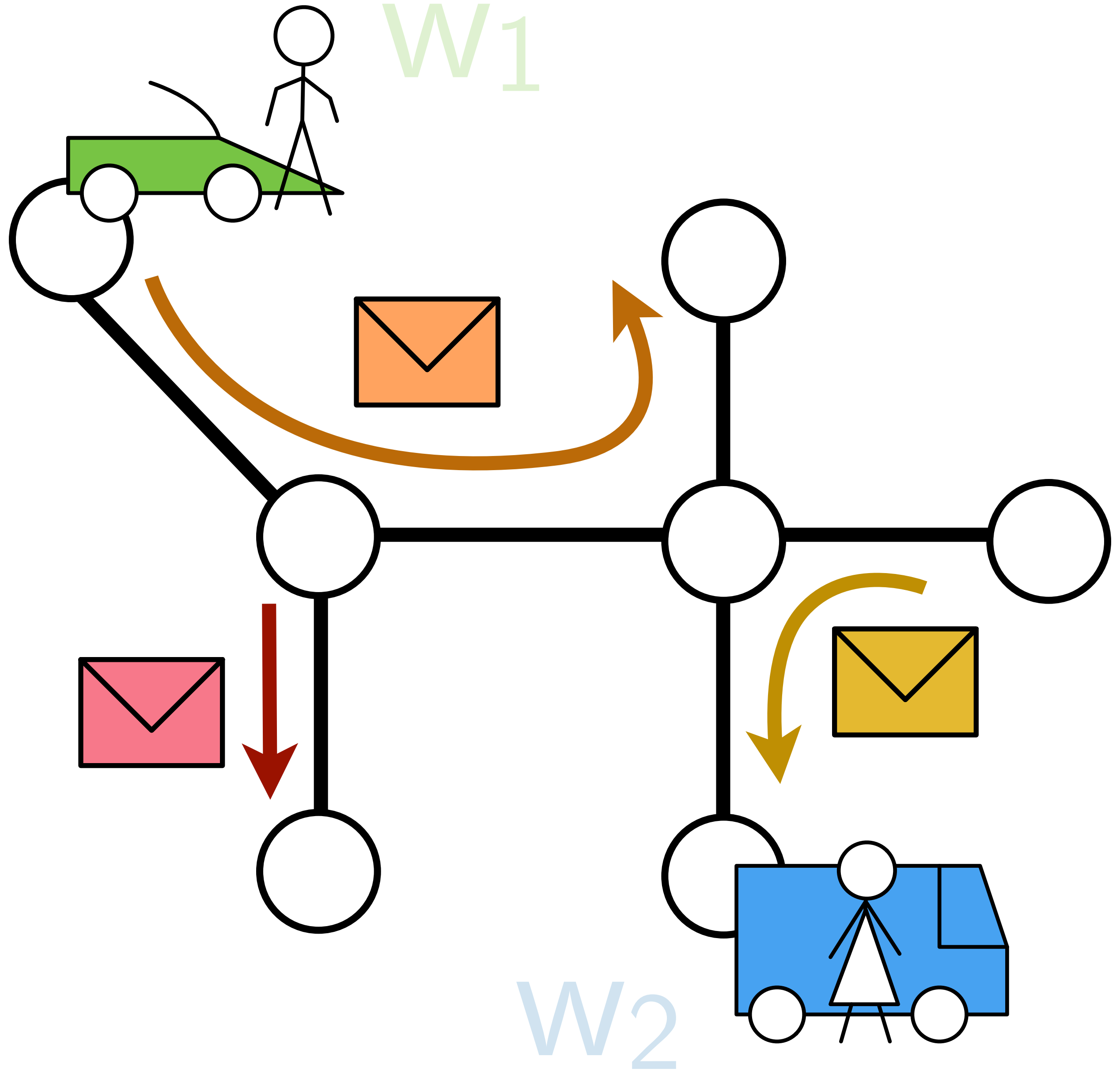
Main Result

Turn existing approximation algorithm into truthful approximation mechanism



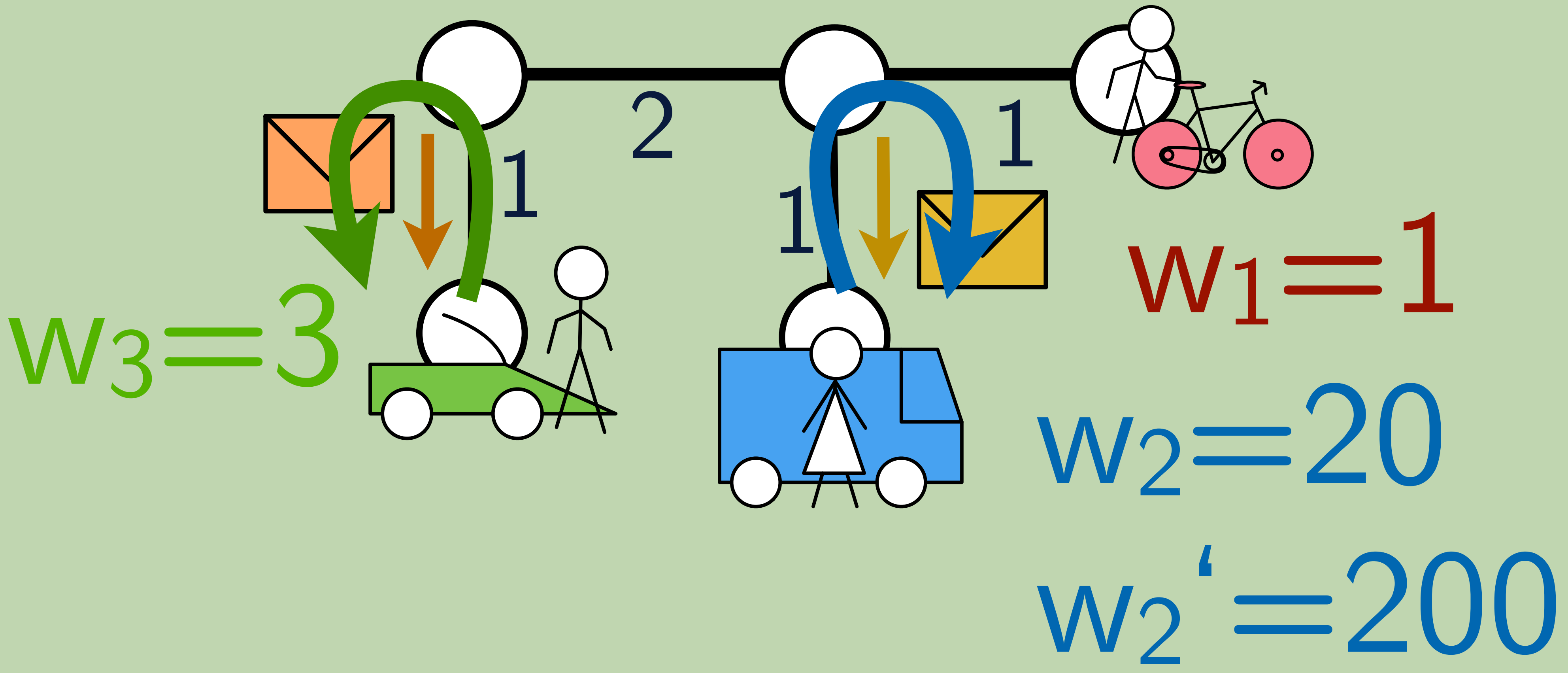
Existing Algorithm A_{pos} [Bärtschi et al.]

- Kruskal-MST-like subgraph search
- $\left(4 \cdot \frac{w_{\max}}{w_{\min}}\right)$ -approximation of cost(OPT)
- weight-independent output schedule



Challenge

Can we just use A_{pos} for a truthful mechanism?



$$\text{cost}(\text{OPT}) = 8$$

$$\text{cost}(A_{pos}) = 46$$

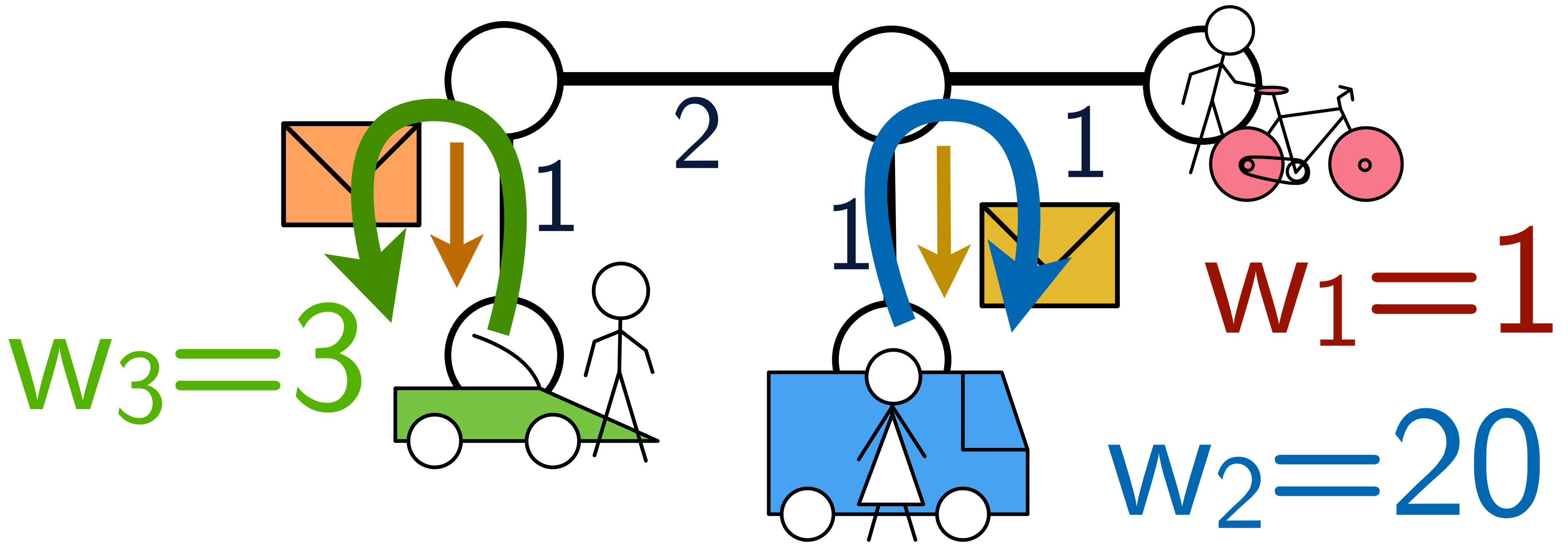
No! Because with any payment rule

- either not truthful
- or no voluntary participation

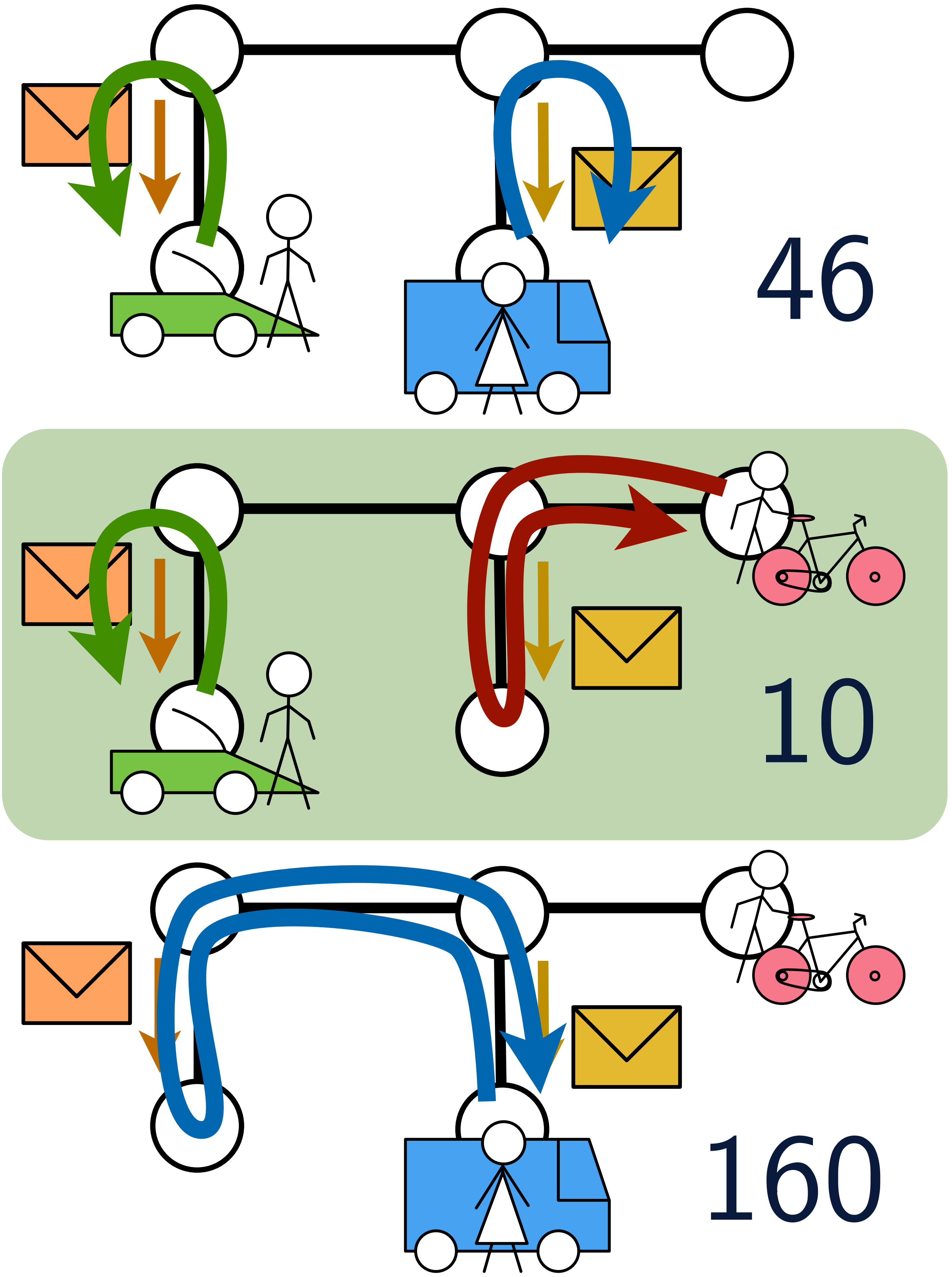
Our Mechanism

Our Approximation Mechanism A*

- run A_{pos} on all subsets of $\geq$k-1 agents
- take cheapest of these k+1 solutions
- use Vickrey-Clark-Groves payments



cost = 46

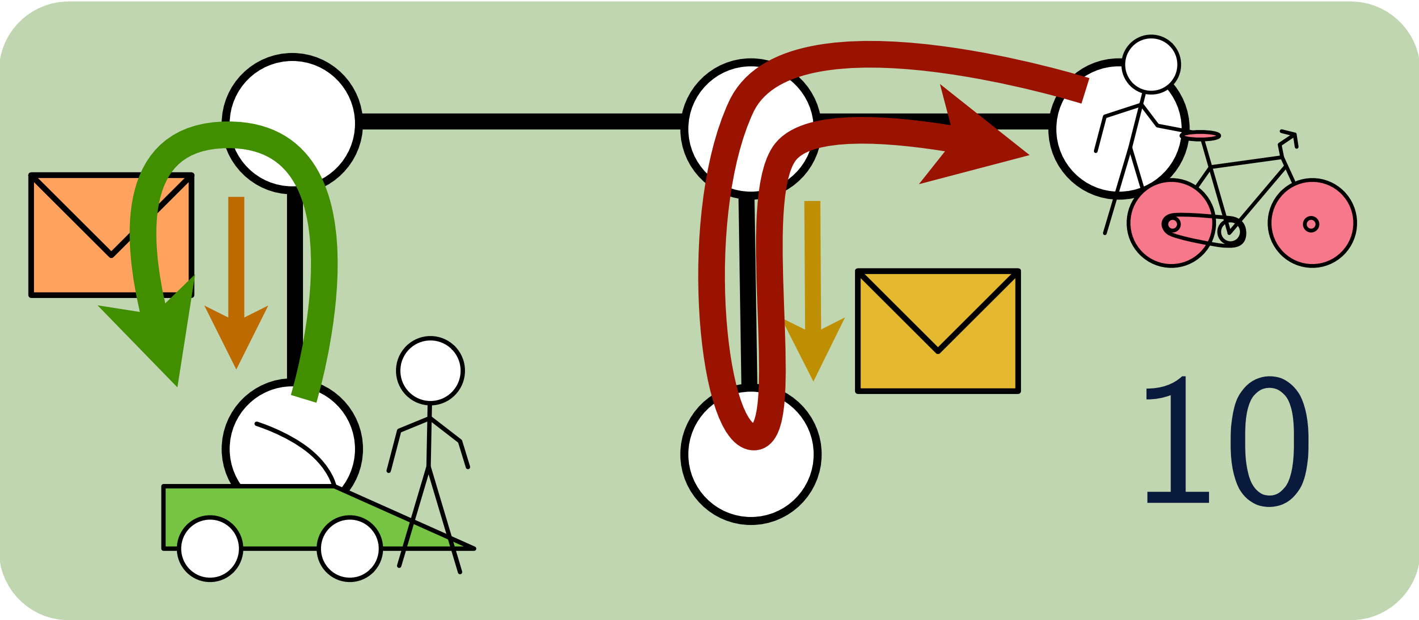


William Vickrey: „Counterspeculation, Auctions and Competitive Sealed Tenders“. Journal of Finance, pages 8–37, 1961.
Edward H. Clarke: „Multipart Pricing of Public Goods“. Public Choice, pages 17–33, 1971.
Theodore Groves: „Incentive in Teams“. Econometrica, 41:617–631, 1973.

Our Approximation Mechanism A*

- run A_{pos} on all subsets of ≥k-1 agents
- take cheapest of these k+1 solutions
- → use Vickrey-Clark-Groves payments

P_1 = 46-6



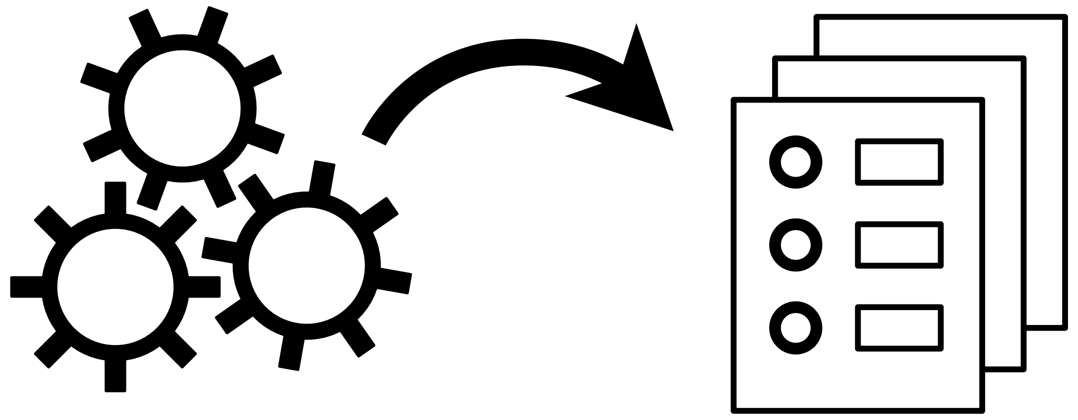
P_3 = 160-4

P_i = (cost of all others with agent i absent) - (cost of all others with agent i present)

William Vickrey: „Counterspeculation, Auctions and Competitive Sealed Tenders“. Journal of Finance, pages 8–37, 1961.
Edward H. Clarke: „Multipart Pricing of Public Goods“. Public Choice, pages 17–33, 1971.
Theodore Groves: „Incentive in Teams“. Econometrica, 41:617–631, 1973.

Our Approximation Mechanism A*

- weight-dependent output schedule



Goals of a good mechanism

🟢	truthfulness	*by VCG payment scheme*
🟢	voluntary participation	*by VCG payment scheme*
🟠	near optimality	*at least as good as* A_{pos}
🟠	frugality	*unclear*
🟢	polynomial running time	$O(\text{poly}(n,m,k))$

Constant Factor

Goals of a good mechanism

- 🟢 truthfulness
- 🟢 voluntary participation
- 🟡 near optimality
- 🟡 frugality
- 🟢 polynomial running time

at least as good as A_{pos}

$\left(4 \cdot \frac{w_{\max}}{w_{\min}}\right)$-approximation

Constant Factor

A^m for constant m

- 2-approximation
- $O(f(m) \cdot \text{poly}(n,m,k))$

🟢 truthfulness
🟢 voluntary participation
🟢 near optimality
🟡 frugality
🟡 polynomial time → FPT

A^k for constant k

- 3.6-approximation
- $O(k^m \cdot poly(n,m,k))$

🟢 truthfulness
🟢 voluntary participation
🟢 near optimality
🟡 frugality
🔴 polynomial time

Constant Factor

Mechanism A^m for constant m (only few packages)

- omit any possible collaboration → 2-approximation
- enumerate all message-to-agent assignments
  $O(m!\cdot(k+m)^m\cdot\text{poly}(n,m,k))$
- optimal assignment via weighted matching
  $O(f(m)\cdot\text{poly}(n,m,k))$ → FPT-algorithm

Constant Factor

Mechanism A^k for constant k (only few agents)

- 368/367-approximation still NP-hard for k=1
- omit any possible collaboration → 2x
- enumerate all $O(k^m)$ partitions
- use stacker crane approximation per agent → 1.8x

→ 3.6-approximation mechanism in $O(k^m \cdot \text{poly}(n,m,k))$

Greg N. Frederickson, Matthew S. Hecht, Chul E. Kim: „Approximation algorithms for some routing problems“. FOCS’76, pages 216–227, 1976.

Frugality

Do we pay much more than cost(OPT)?

Goals of a good mechanism

- ● truthfulness
- ● voluntary participation
- ● near optimality
- ● frugality *unclear*
- ● polynomial running time

Frugality

Do we pay much more than cost(OPT)?

- requirement: *monopoly freedom*
  (some optimum solution uses multiple agents)

- for a single package (m=1), we can show
  - mechanism A_{OPT} pays at most 2·cost(OPT)
  - mechanism A^m pays at most 2.88·cost(OPT)

Key Points

model: cargo company who hires selfish drivers

turn approximation algorithm into mechanism

improve the guarantees for certain special cases

Open Problem:

polynomial time constant factor approximation algorithm