Algorithms for Distributed Team of *Energy-Constrained* Mobile Robots

Shantanu Das

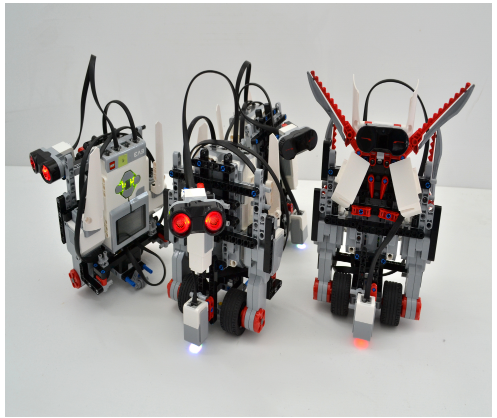
Aix-Marseille University, France

Joint work with:

Andreas Bärtschi; Jérémie Chalopin; Yann Disser; Barbara Geissmann; Daniel Graf;

Jan Hackfeld; Arnaud Labourel; Matus Mihalak; Paolo Penna; Peter Widmayer

Distributed Network of Robots



Distributed System of robots

> That move autonomously
> Perform sensing, data gathering, exploration
> Fault Tolerant, self adjusting

Highly Dynamic Networks, Mobile Sensor Networks

Large Teams of Small Robots

Limitations of Robots ⟷ Distributed Tasks

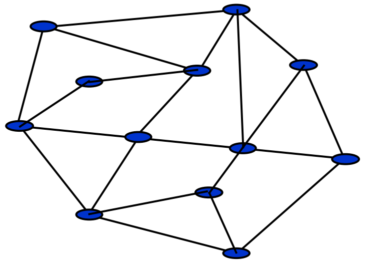
Small and inexpensive robots

- Limited Memory
- Limited Visibility
- Limited communication
- Inaccurate measurement

Major Issue: Energy consumption

Moving consumes Energy

Limited Energy ⟷ **Distributed Tasks**

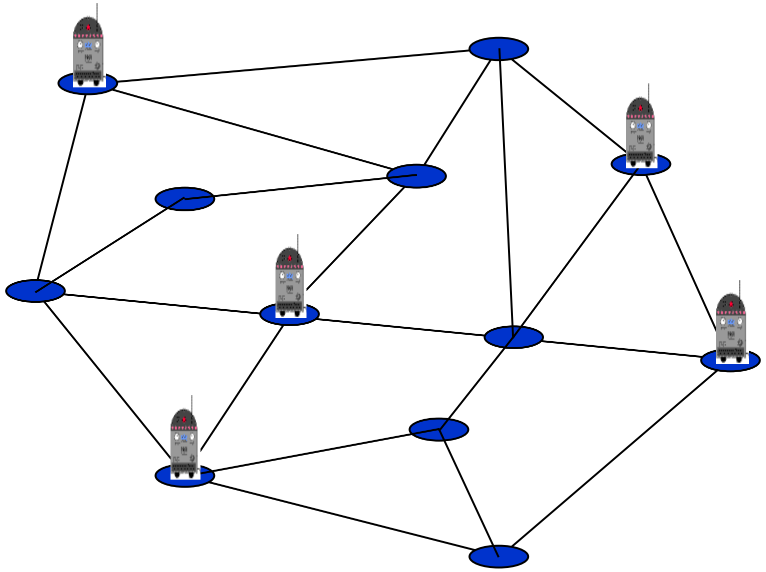


- Moving consumes more energy than computing!
- Small robots cannot have ***a large Fuel-Tank (or Battery)!***
- Robots cannot refuel or recharge while moving!

<u>Assumption</u>:
[Energy bound = B]   => At most B moves per robot.

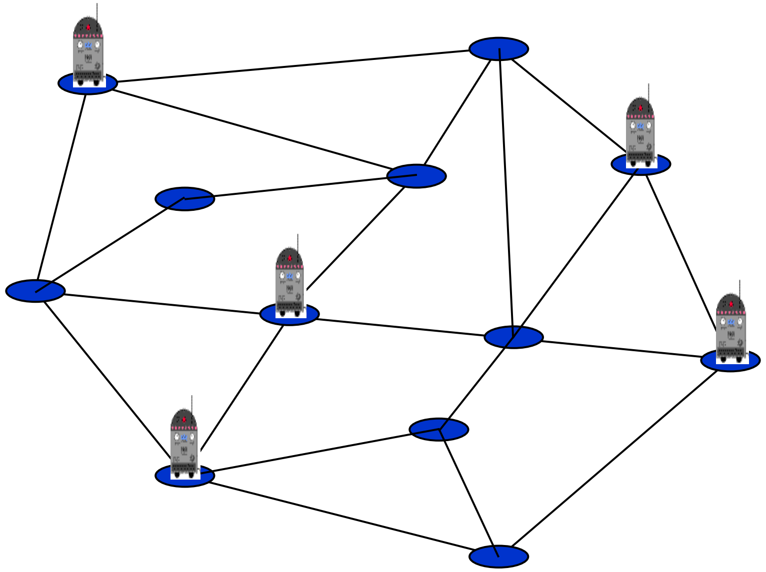
The Model

- Undirected Weighted Graph G
- Robots start from a set of nodes
- Each robot can travel a distance of at most **B**



The Problems

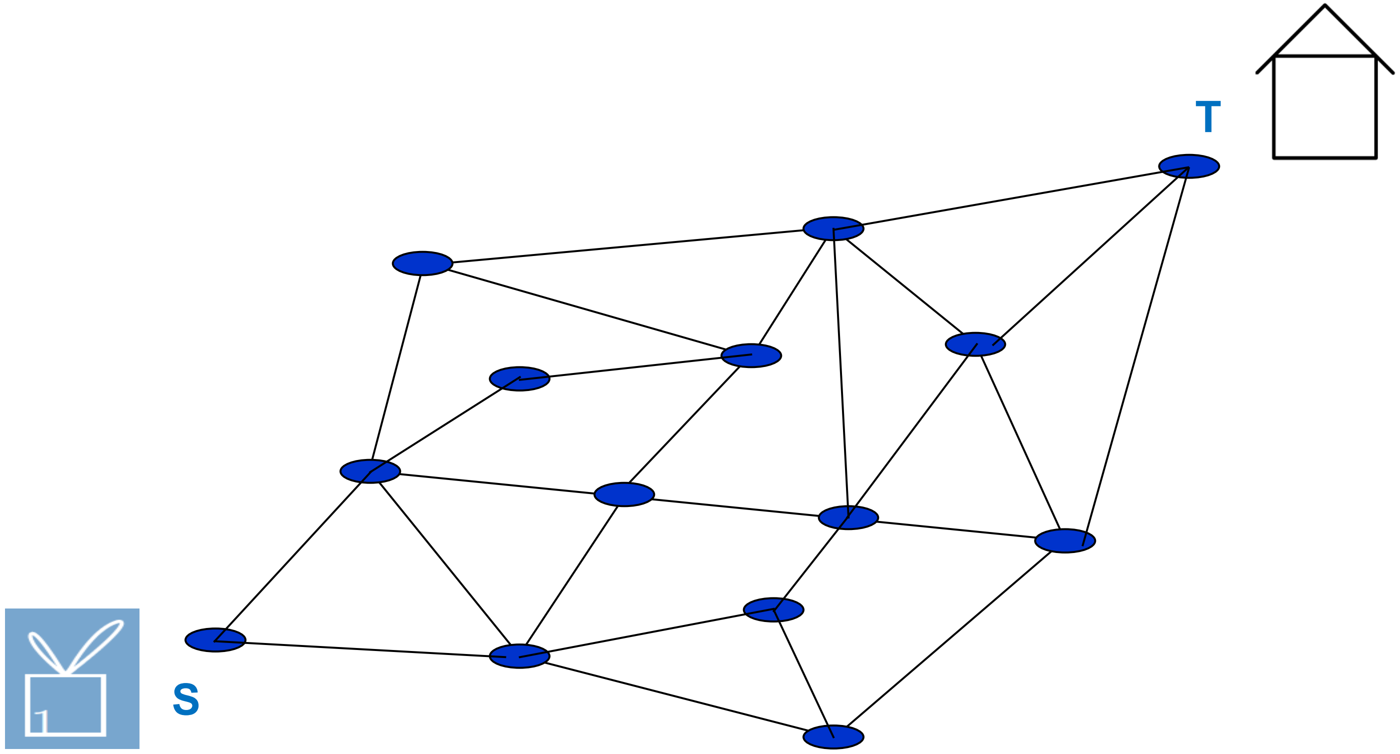
- Undirected Weighted Graph G
- Robots start from a set of nodes
- Each robot can travel a distance of at most **B**



- *Exploration / Search*
- Delivering Goods
- *Patrolling / Guarding*
- *Rendezvous*
- *Pattern Formation*

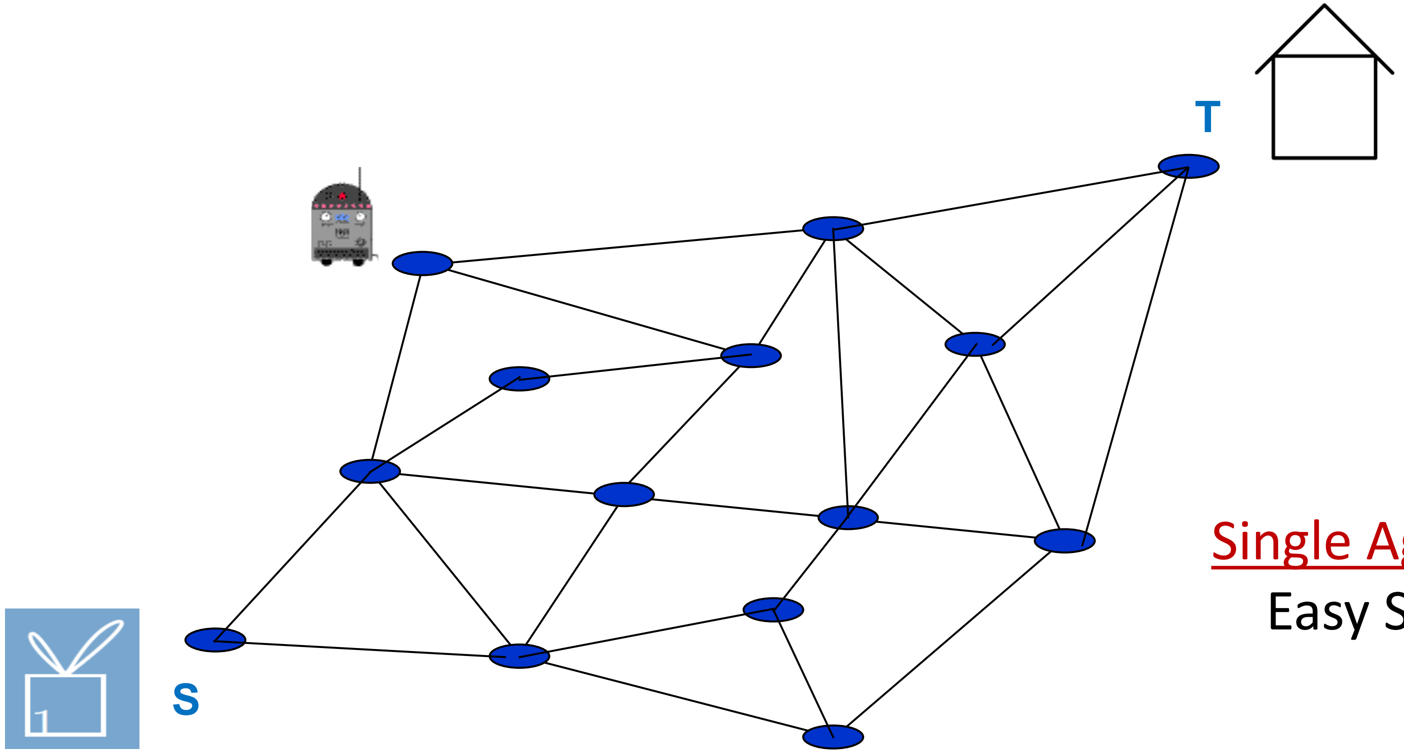
A simple Problem : Delivery

- Move an item from source to target



A simple Problem : Delivery

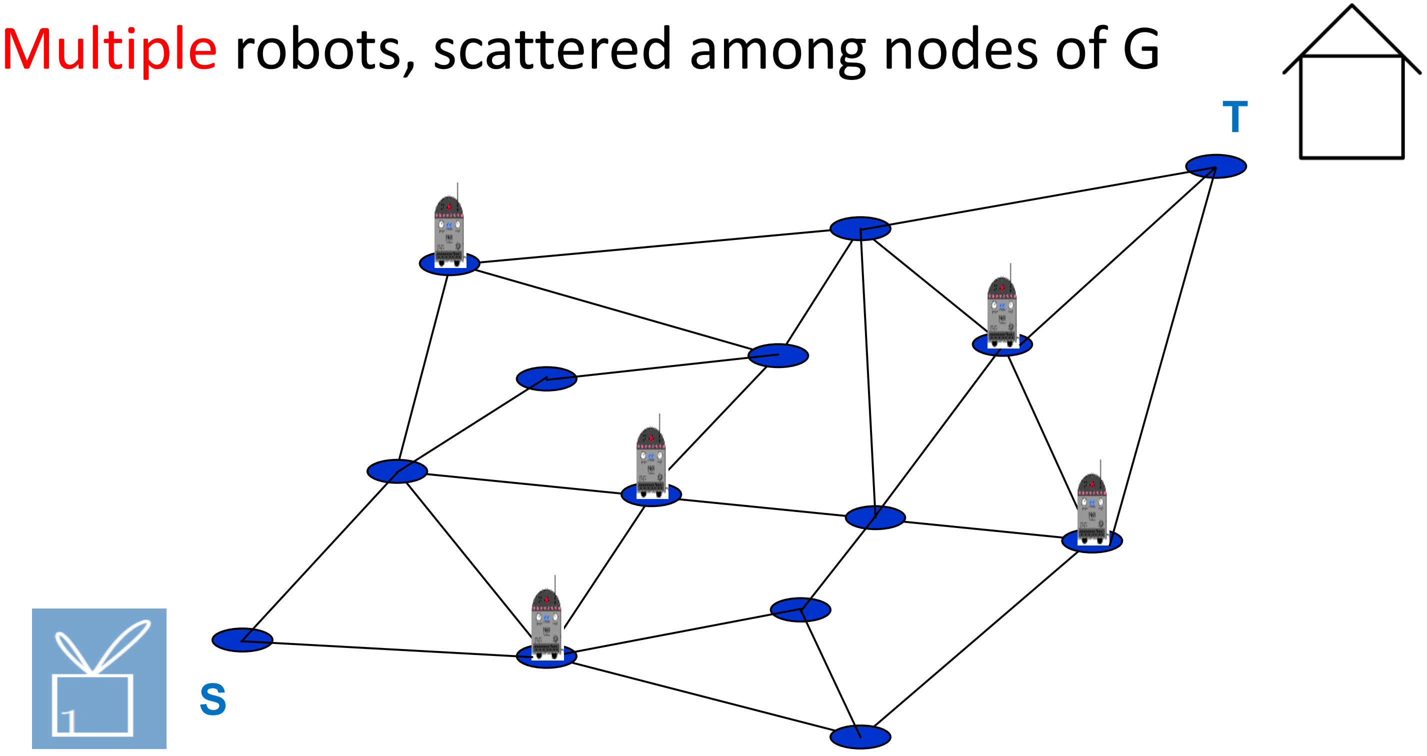
- Move an item from source to target



Single Agent:
Easy Solution

A simple Problem : Delivery

- Move an item from source to target
- Multiple robots, scattered among nodes of G

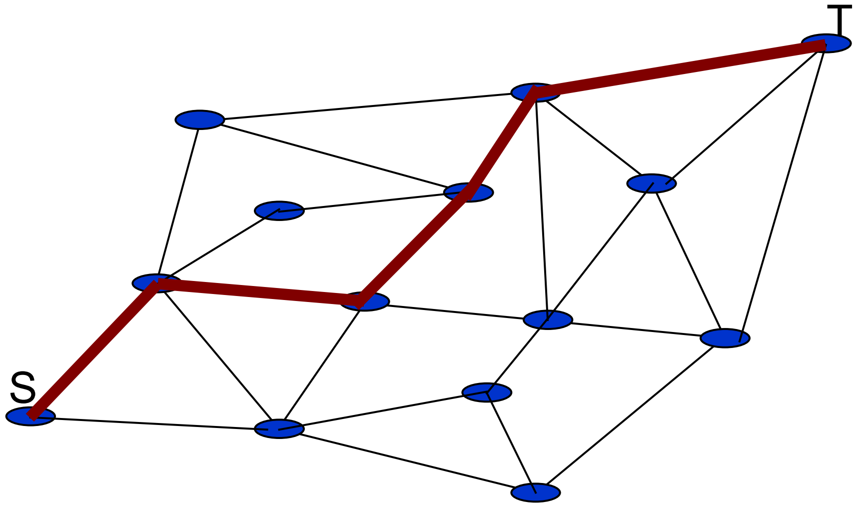


Collaborative Delivery

<u>Definition</u>: Given **G**(V,E); **w**; **s**,**t** $\in$ V; $p_1, p_2, \ldots p_k \in$ V; **B**

Q: Is there a schedule for robots starting at $p_1, p_2, \ldots p_k$, such that no robot moves more than B and the item is delivered from s to t?

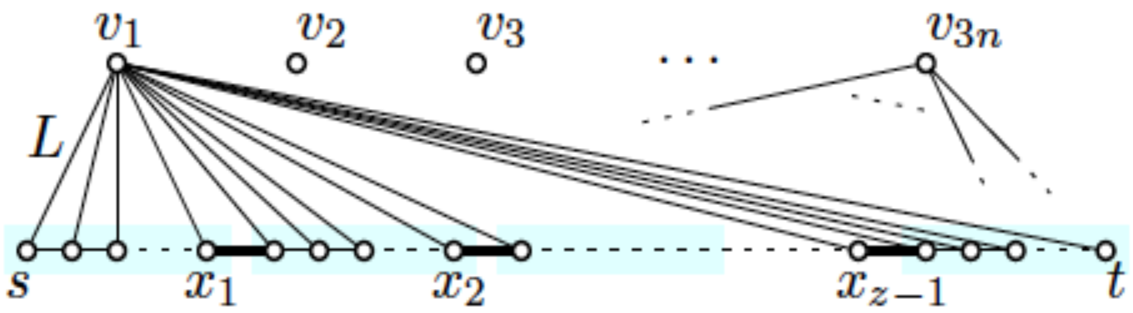
- The item is moved along a simple s-t path P (not shortest)
- Each robot pushes the item on a continuous segment of P.



Complexity of Delivery

THEOREM: Collaborative Delivery is NP-complete for arbitrary graph with many agents.

- By a reduction from 3-PARTITION Problem



3-Partition:
S={a_1,a_2, … a_3m}
Find S_1,S_2,…S_m
s.t. Sum(S_i) = B
& |S_i| = 3

Easy instances of Delivery

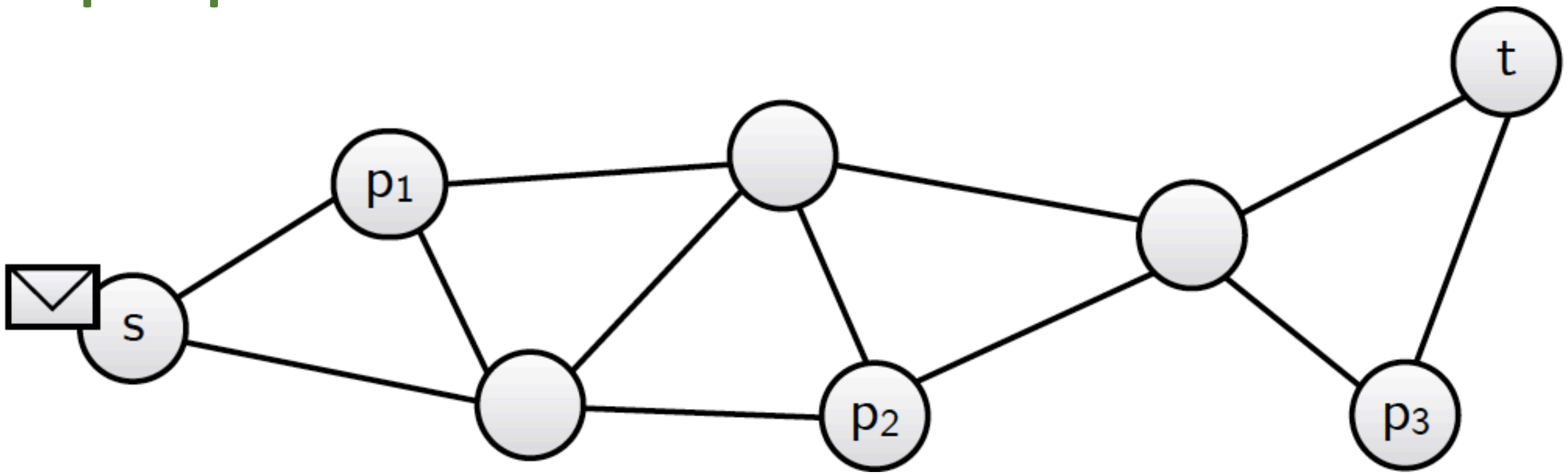
When is it easy to solve Collaborative Delivery?

• **If the delivery path P is fixed?**	**NO**
❖ **If the order of robots is fixed?**	**?**
• **If the number of robots is constant?**	
• **If the energy budgets are constants?**	
• **Specific graphs:** • **Planar graphs** • **Trees**	

Delivery with fixed order

THEOREM: When the order on robots is fixed, Collaborative Delivery can be solved in time $O(k(n+m)(n \cdot \log n + m))$

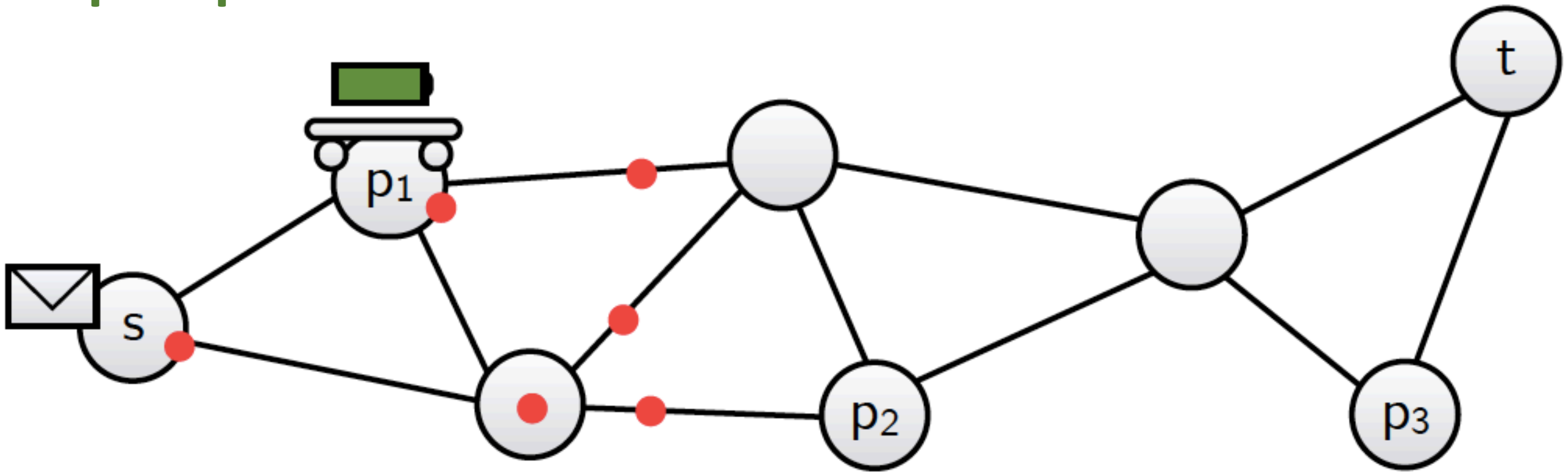
p1 < p2 < p3



Delivery with fixed order

THEOREM: When the order on robots is fixed, Collaborative Delivery can be solved in time $O(k(n+m)(n \cdot \log n + m))$

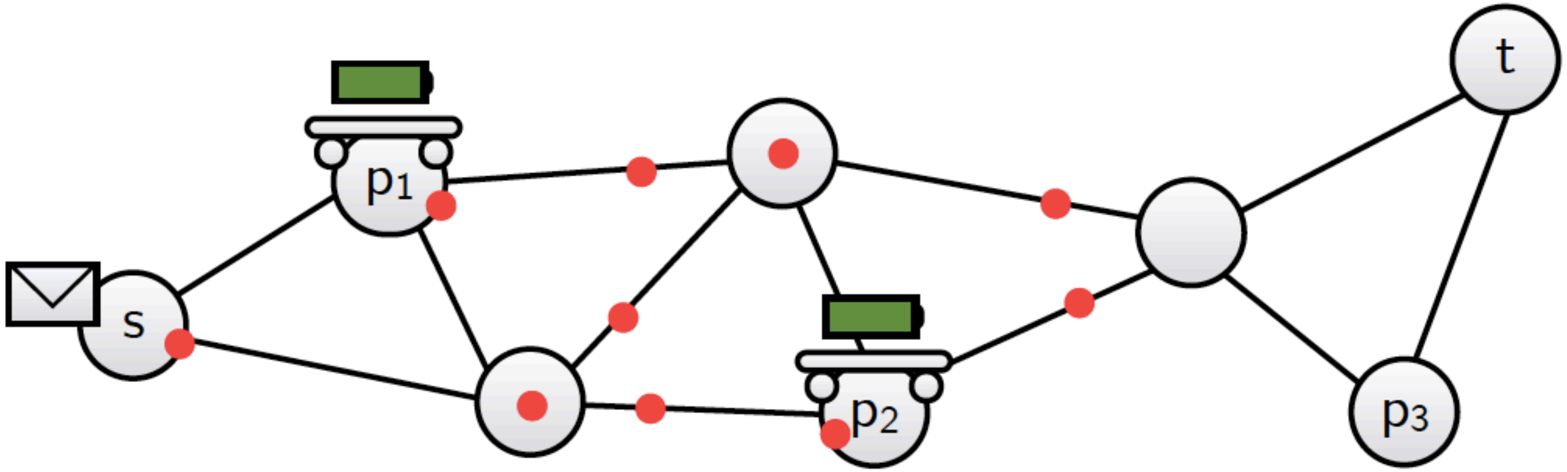
p1 < p2 < p3



Delivery with fixed order

THEOREM: When the order on robots is fixed, Collaborative Delivery can be solved in time $O(k(n+m)(n \cdot \log n + m))$

p1 < p2 < p3



Delivery with fixed order

THEOREM: When the order on robots is fixed, Collaborative Delivery can be solved in time $O(k(n+m)(n \cdot \log n + m))$

COROLLARY:
For a constant number of robots k, Collaborative Delivery can be solved in polynomial time

Algorithm: Brute force, trying all possible ordering of robots

Easy instances of Delivery

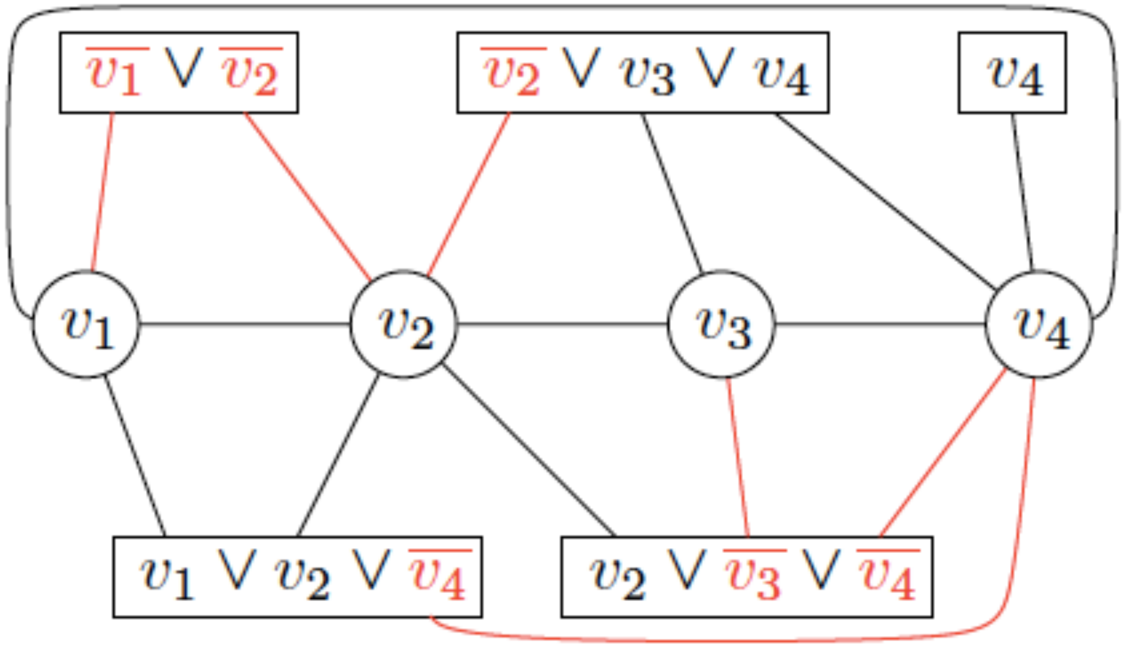
When is it easy to solve Collaborative Delivery?

• **If the delivery path P is fixed?**	**NO**
• **If the order of robots is fixed?**	**YES**
• **If the number of robots is constant?**	**YES**
• **If the energy budgets are constants?**	**NO**
• **Specific graphs:** ➢ **Planar graphs** ➢ **Trees**	**?**

Planar Graphs

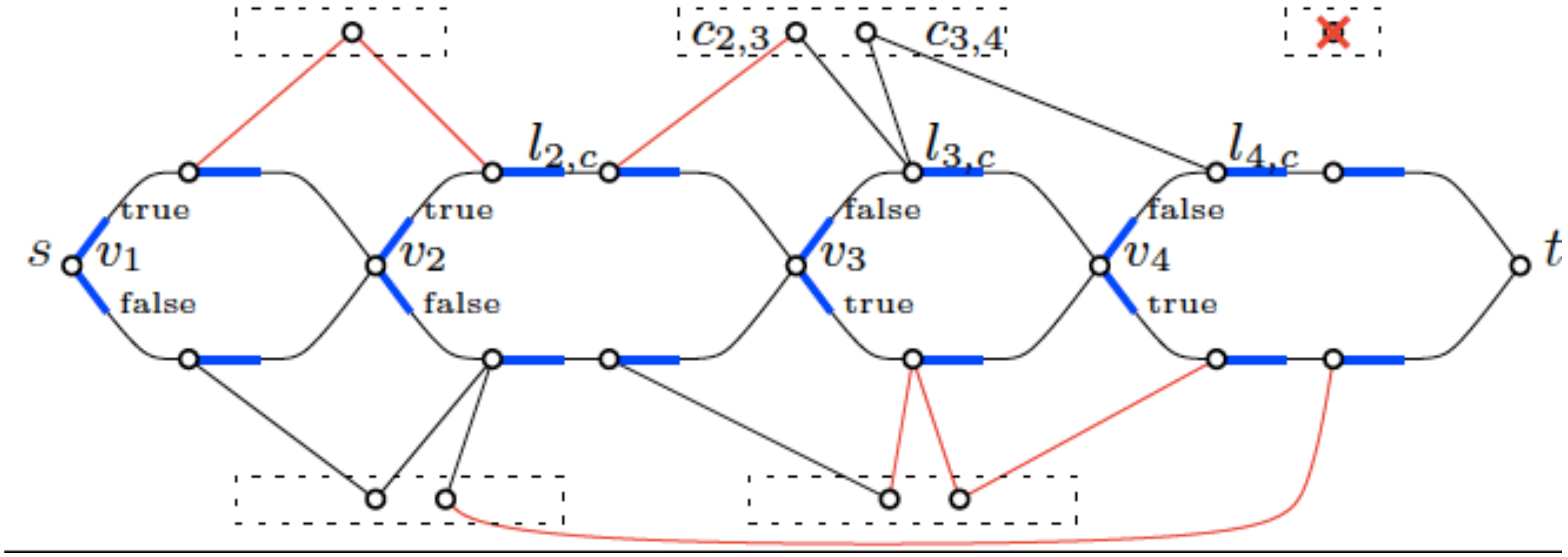
THEOREM: Collaborative Delivery is NP-complete even in Planar graphs.

Planar 3-SAT Instance



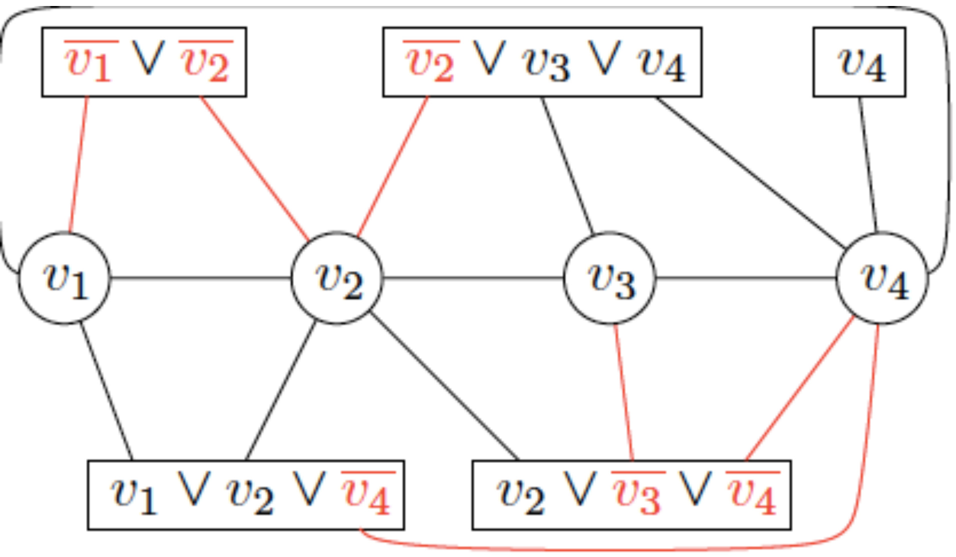
Planar Graphs

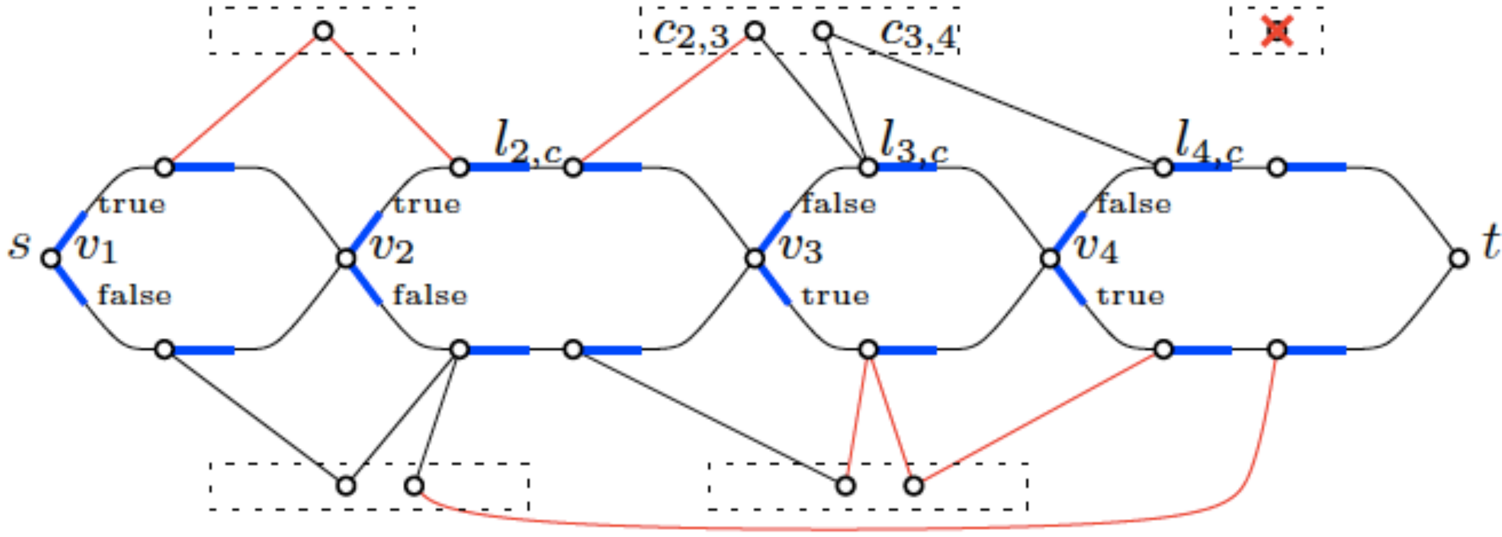
THEOREM: Collaborative Delivery is NP-complete even in Planar graphs.



Planar Graphs

THEOREM: Collaborative Delivery is NP-complete even in Planar graphs.





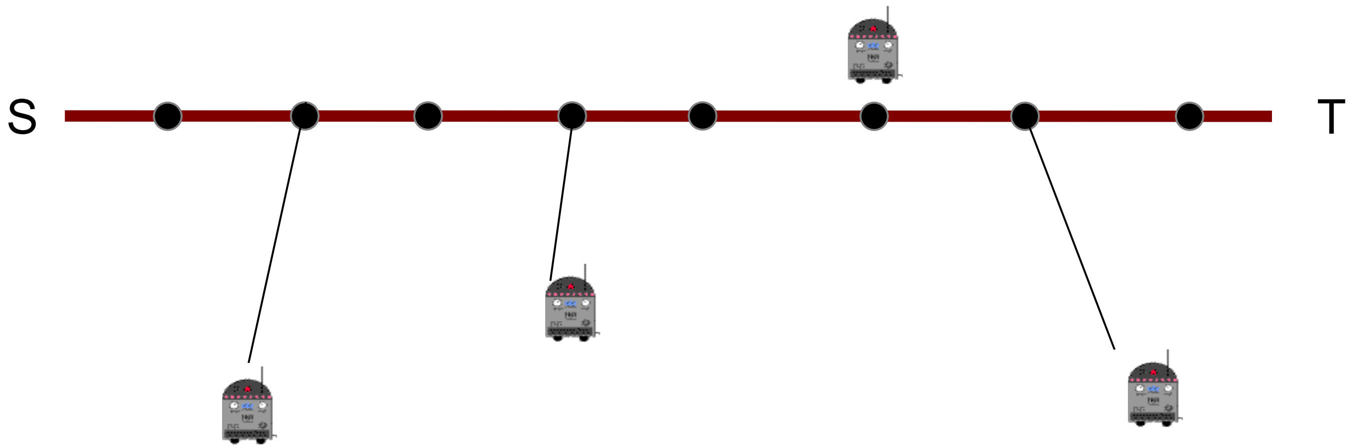
Easy instances of Delivery

When is it easy to solve Collaborative Delivery?

• If the delivery path P is fixed?	NO
If the order of robots is fixed?	YES
• If the number of robots is constant?	YES
• If the energy budgets are constants?	NO
• Specific graphs: ➢ Planar graphs ➢ Trees	NO ?

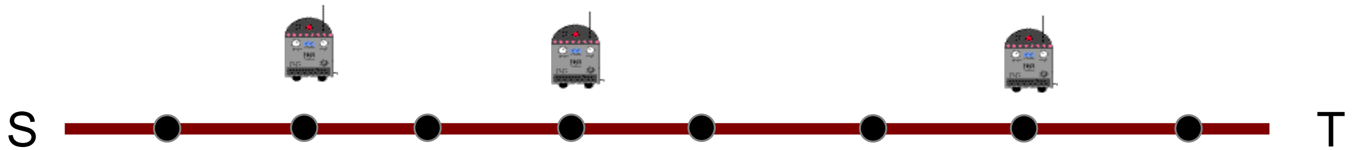
Collaborative Delivery on a Tree

- There is a *unique* s-t path P.
- Each robot has a unique path to reach P.
- The problem reduces to a path



Collaborative Delivery on a Line

- Collaborative Delivery on a line is poly-time solvable, if each robot is already on the line and has same energy B.



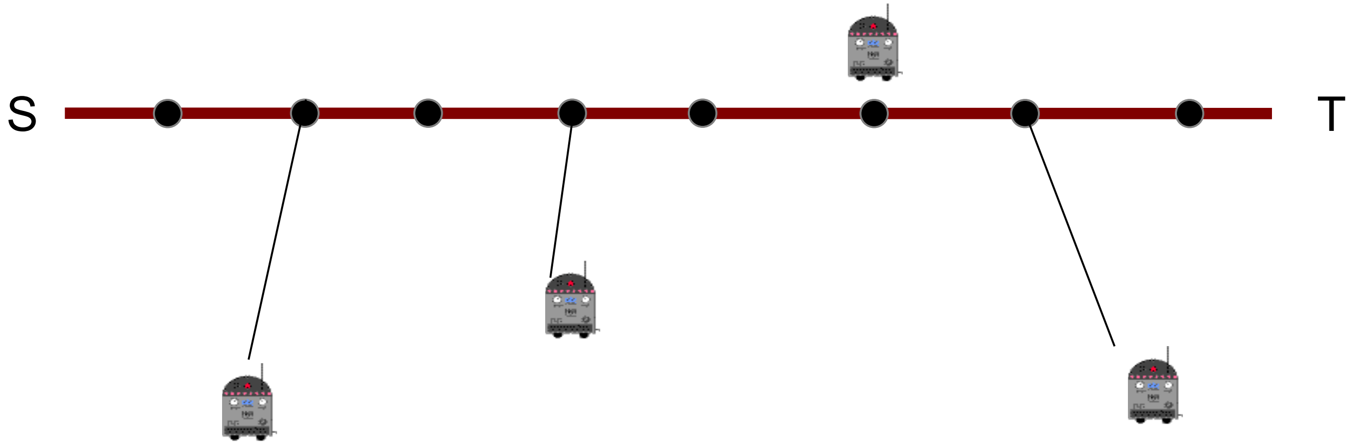
If robots have arbitrary energy levels (B_1,B_2,B_3,B_4 ...) ?

- Collaborative-Delivery on a line is (weakly) NP-hard !
- Reduction from Weighted-4-partition problem.
  [Chalopin et al. ICALP 2014]

Collaborative Delivery on a Tree

<u>THEOREM</u>: Collaborative Delivery is weakly NP-complete for trees with many agents.

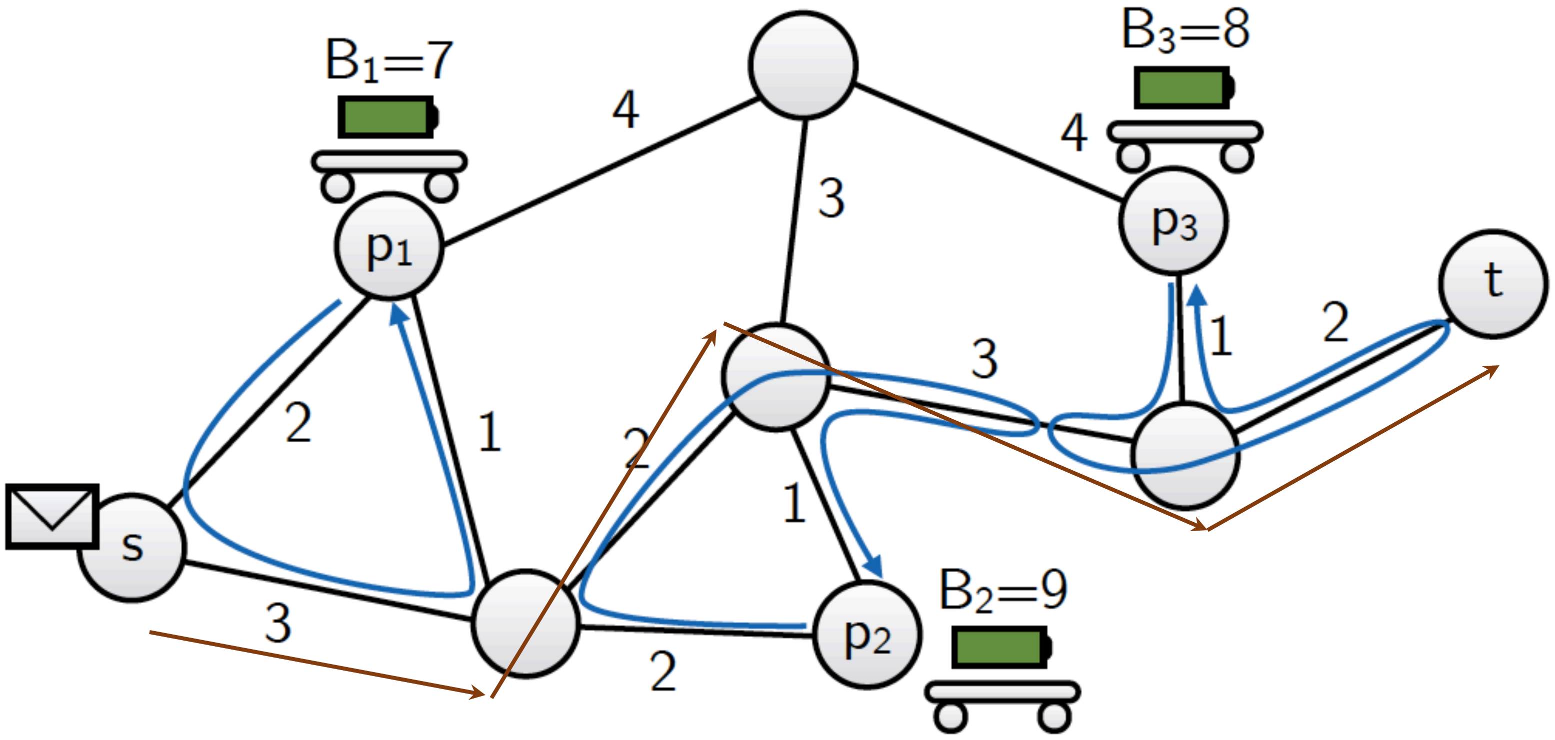
- There exists pseudo-polynomial algorithm (dynamic programming)



Collaborative Delivery with Returning Agents

Delivery with Return

- Each agent needs to return to home station.



Delivery with Return

Definition: Given **G**(V,E); **w**; **s,t** $\in$ V; $p_1,p_2,\ldots p_k \in V$; **B**

Q: Is there a schedule for robots starting at $p_1,p_2,\ldots p_k$, such that

- each robot performs a tour of length B (returning home)
- the item is delivered from s to t?

- What is the complexity of Collaborative Delivery with Return?

Delivery with Return

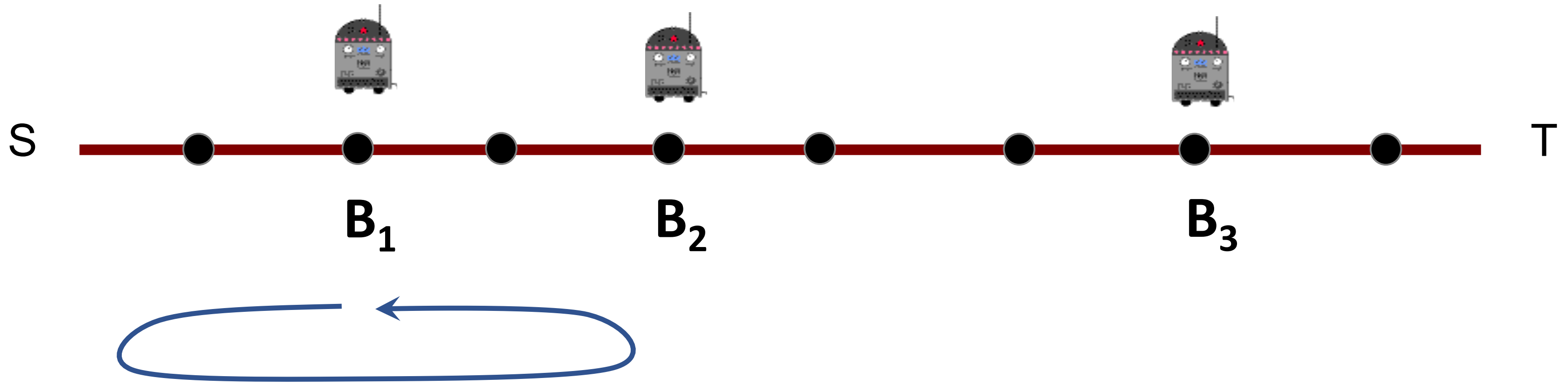
- Collaborative Delivery with Return for many agents is NP-hard, even in Planar graphs
- Collaborative Delivery with Return can be solved in polynomial time for constant number of agents (or if the order of agents is given)

THEOREM: ***Collaborative Delivery with Return*** can be solved in polynomial time in Trees

Recall: Collaborative Delivery (without Return) is NP-complete in Trees!

Delivery with Return in Trees

- Reduce the problem to the Path

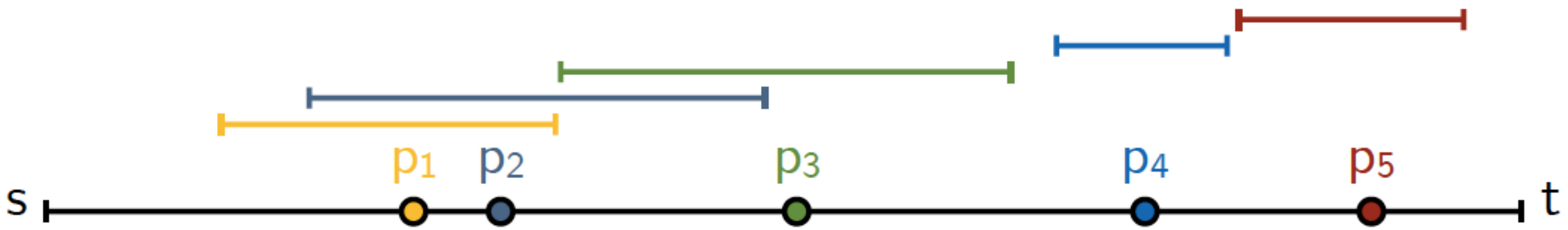


Delivery with Return in Trees

- Reduce the problem to the Path.
- Each agent corresponds to an interval of fixed length.

Delivery with Return in Trees

- Reduce the problem to the Path
- Each agent corresponds to an interval of fixed length.
- Cover the line with intervals (choose greedily)
- O(n + k log k) algorithm for delivery in Trees using k agents.



Variants of Collaborative Delivery

• If the delivery path P is fixed	Hard
• If the number of robots is constant (or order is fixed)	EASY
• If the energy budgets are constants	Hard
• Specific graphs: ➢ Planar graphs ➢ Delivery (without Return) on Trees ➢ Delivery with Return on Trees	 Hard Hard EASY

Approximations

Optimization of Energy Budget B

Approximation Algorithm: Algorithm solves delivery using c.B energy if the optimal solution uses B energy per agent.

There is a polynomial time 2-approximation algorithm

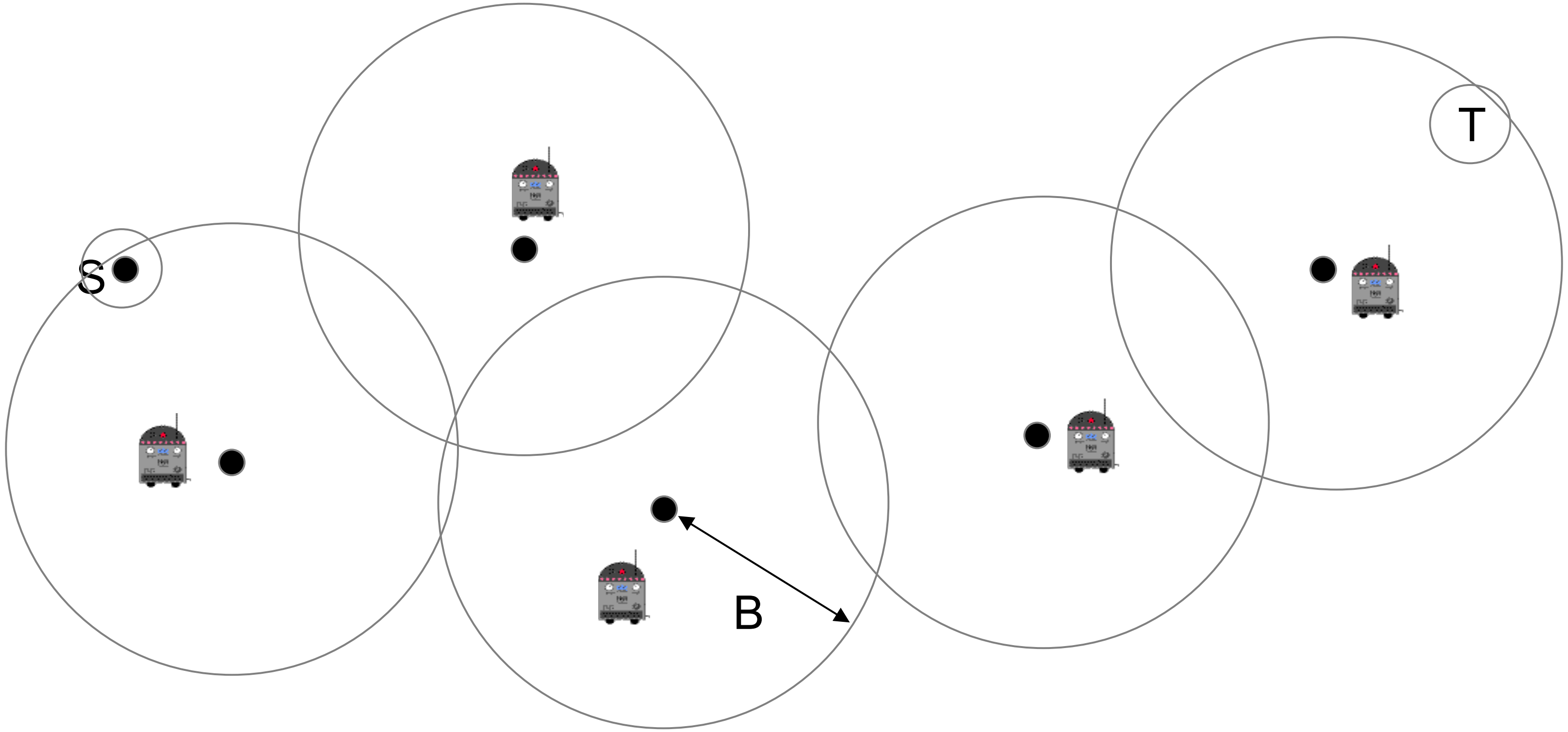
Distinct Energy Budgets

Resource-augmented Algorithm: Algorithm solves delivery using c.B(i) energy for robot r(i) whenever there is a solution to the original instance of the problem.

There is a polynomial time 3-resource augmented algorithm

Algorithm for Delivery

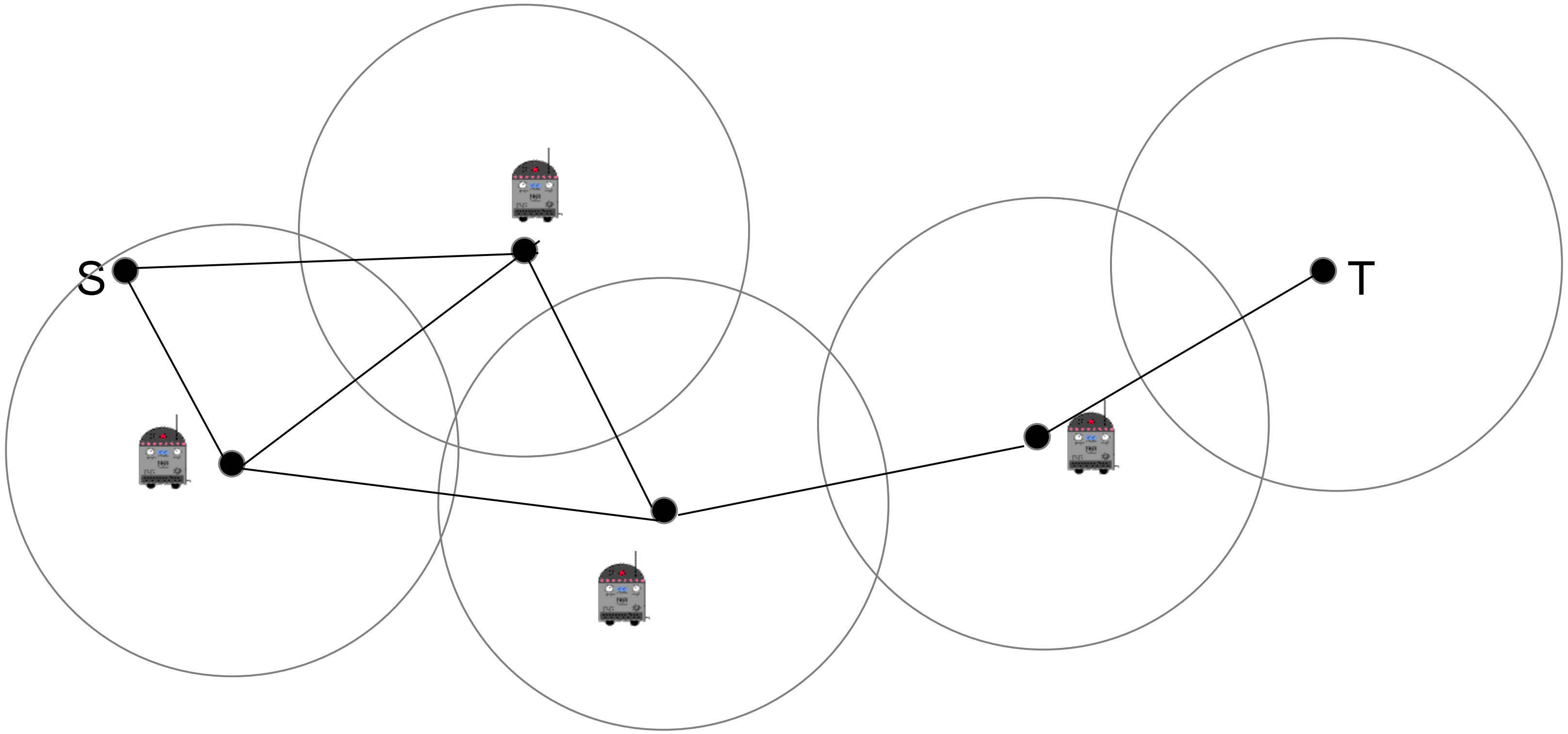
Necessary Condition:



Algorithm for Delivery

Necessary Condition:

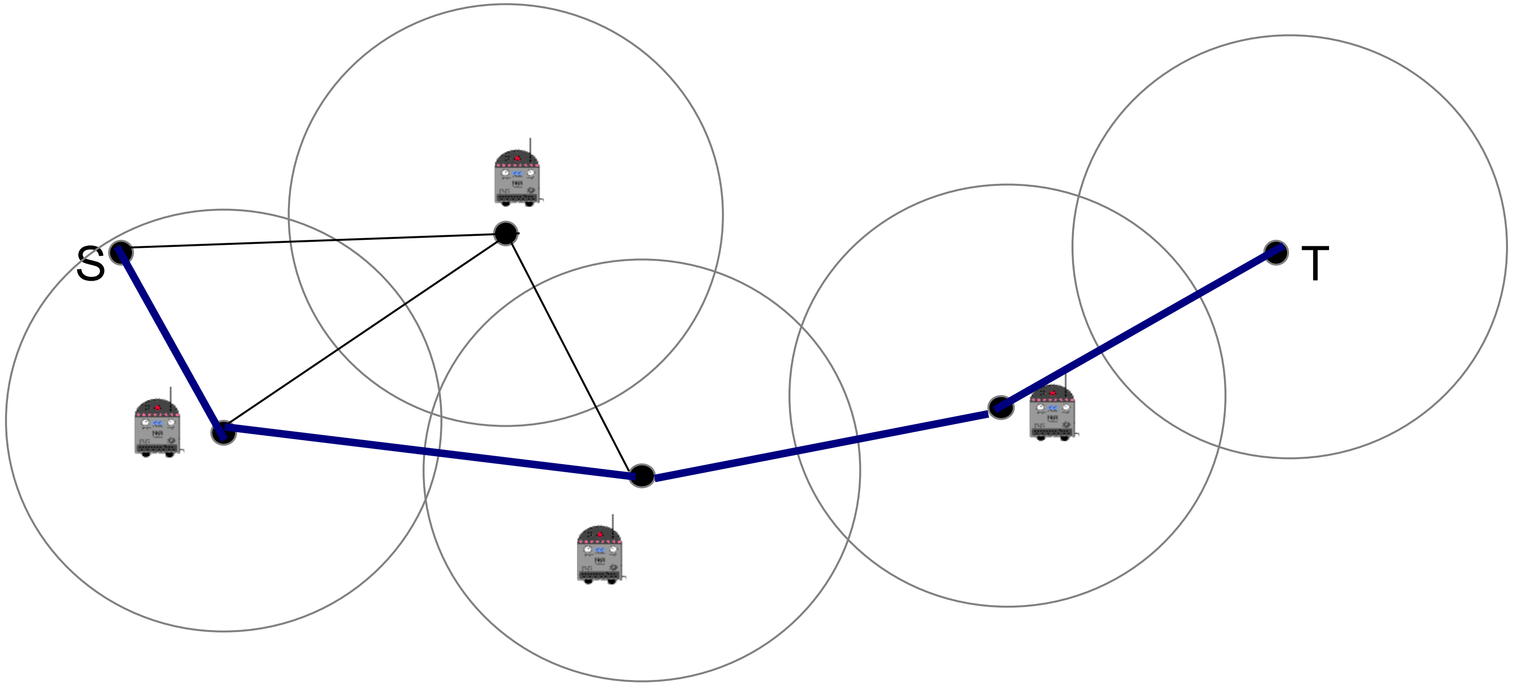
- There exists a S-T path in the intersection graph.



Algorithm for Delivery

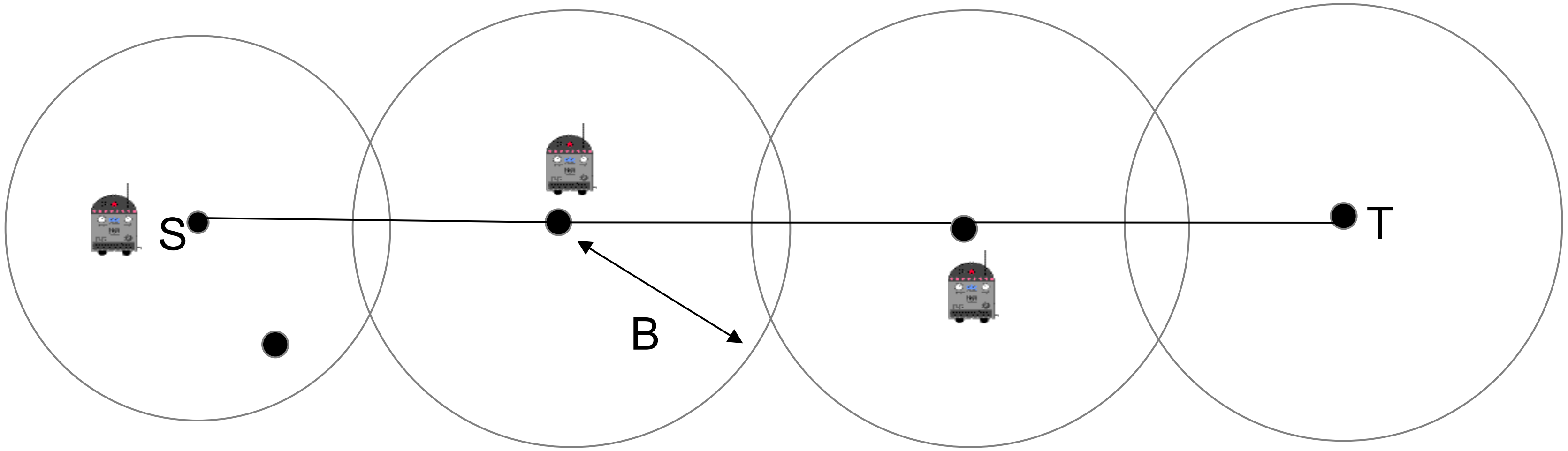
If there exists a S-T path in the intersection graph,

$\Rightarrow$ there is poly-time algorithm using 3B(i) energy for robot r(i).



2-Approx. Algorithm

- Guess the first robot r_1 in the optimal strategy.
- Place r_1 at S with reduced energy (smaller ball).
- Each robot can carry to neighboring robot using 2B energy.



Approximate Algorithms

Optimization version

- **There is a polynomial time 2-approximation algorithm**

Distinct Budget version

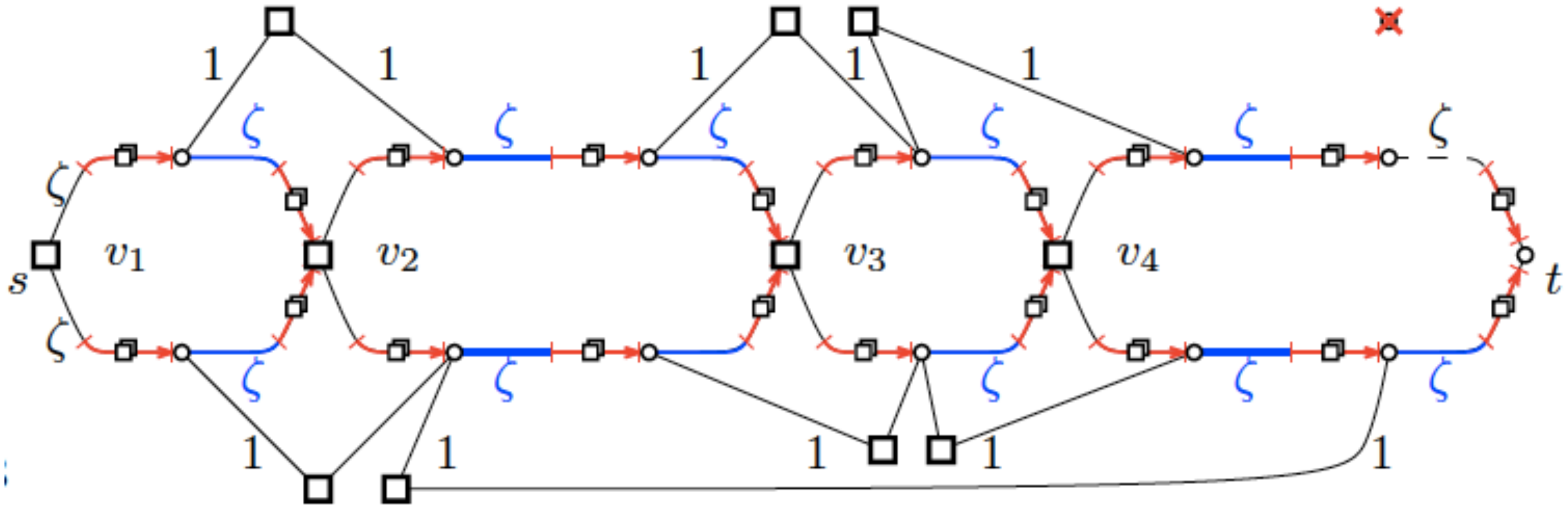
- **There is a polynomial time 3-resource augmented (2-resource augmented) algorithm for collaborative delivery (with return)**

Inapproximability

Theorem: There is no polynomial-time $(2-\epsilon)$-resource-augmented, resp. $(3-\epsilon)$-resource augmented, algorithm for delivery with return (resp. without return) unless P=NP

In-Approximability

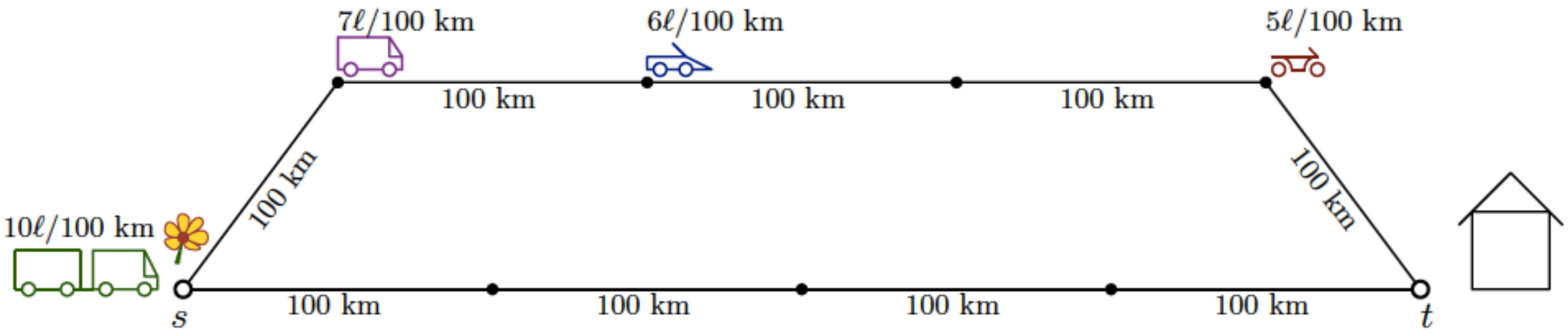
<u>Theorem</u>: There is no polynomial-time (2-ϵ)-resource-augmented, resp. (3-ϵ)-resource augmented, algorithm for delivery with return (resp. without return) unless P=NP



Collaborative Delivery with Heterogeneous Agents

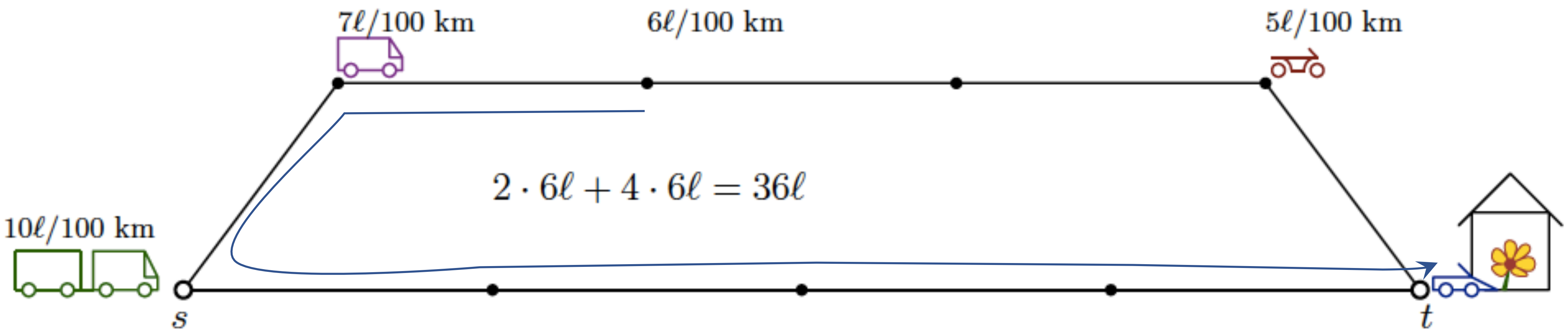
Heterogeneous Agents

Agents differ in their rate of energy consumption …



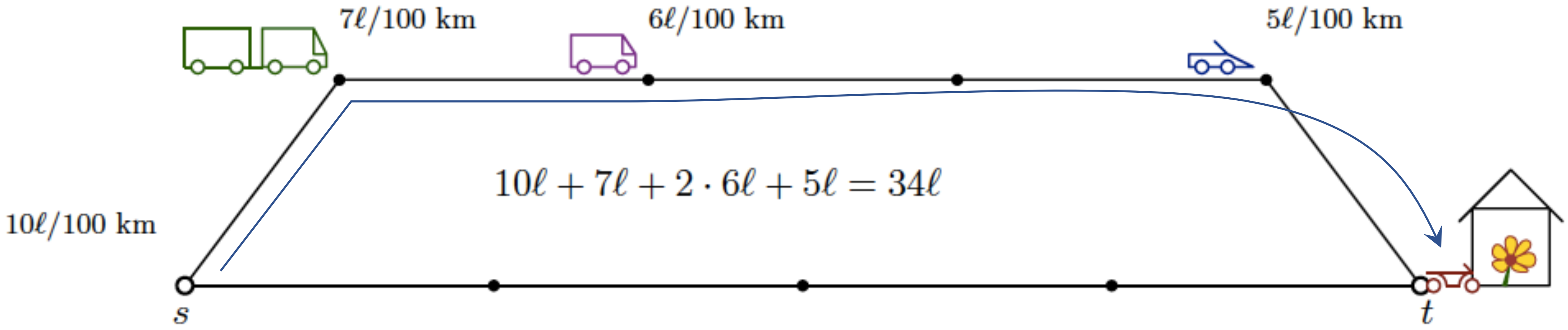
Heterogeneous Agents

Agents differ in their rate of energy consumption …



Heterogeneous Agents

Agents differ in their rate of energy consumption …



Heterogeneous Agents

Collaborative Delivery with Heterogenous Agents:

- k agents start at $\mathbf{p_1,p_2,\ldots\ p_k}$, and
- Have energy consumption rate $\mathbf{w_1,w_2,\ldots\ w_k}$,
- **m** packages to be delivered between (s_1,t_1) (s_2,t_2) … (s_m,t_m)

Objective: Minimize Cost = SUM ($\mathbf{w_i}$. $\mathbf{D_i}$)

where $\mathbf{D_i}$ is the total distance traveled by i-th agent.

Heterogeneous Agents

Single Message:

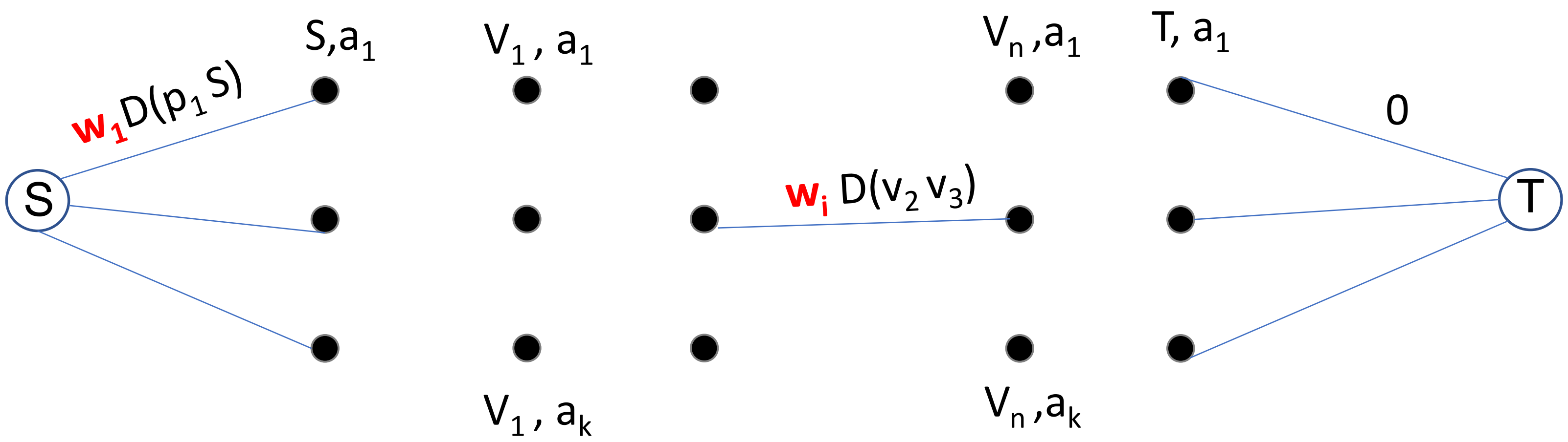
Lemma: If the message is delivered by $p_1, p_2, \ldots p_k$ in this order, then

$$\mathbf{w_1 > w_2 > \ldots > w_k}$$

Theorem: The optimal solution for single message delivery can be computed in O(n^3) time.

Heterogeneous Agents

<u>Theorem</u>: The optimal solution for single message delivery can be computed in O(n^3) time.



Shortest s-t path in auxiliary graph Gx => Optimal Solution

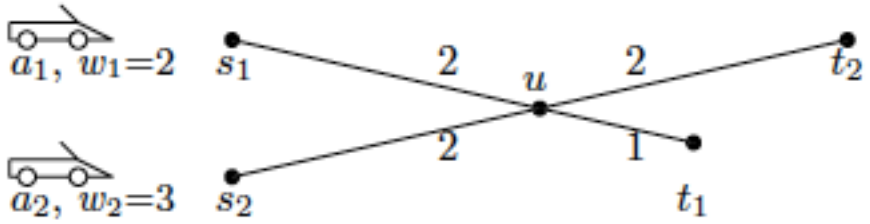
Heterogeneous Agents

Single Message:

Lemma: If the message is delivered by $p_1,p_2,\ldots p_k$ in this order, then

$$\mathbf{w_1 > w_2 > \ldots > w_k}$$

Many Messages:



Delivery of Multiple Packages

Delivery with Heterogenous Agents:

- **Collaboration:**

Should the agents collaborate on delivering a package?

- **Planning:**

How to plan the route of an agent delivering multiple packages?

- **Coordination:**

How to assign messages to agents?

Collaboration

What is the benefit of Collaboration?

- Delivery without **Collaboration:**

Algorithms where each package is delivered by single agent : $\mathbf{Algo_S}$

Definition**:**

$$\mathbf{BoC} = \text{MIN} [\ \text{Cost}\ (\mathbf{Algo_S})\ /\ \text{Optimal Cost}\ ]$$

Benefit of Collaboration

Lower Bound (BoC >= 2)

Any algorithm for delivery *without collaboration* cannot achieve an approximation ratio better than 2

For a single message, the ratio cannot de better than 1/ln(2) =1.44

Upper Bound (BoC <= 2)

There is an algorithm for delivery *without collaboration* that has approximation ratio <= 2

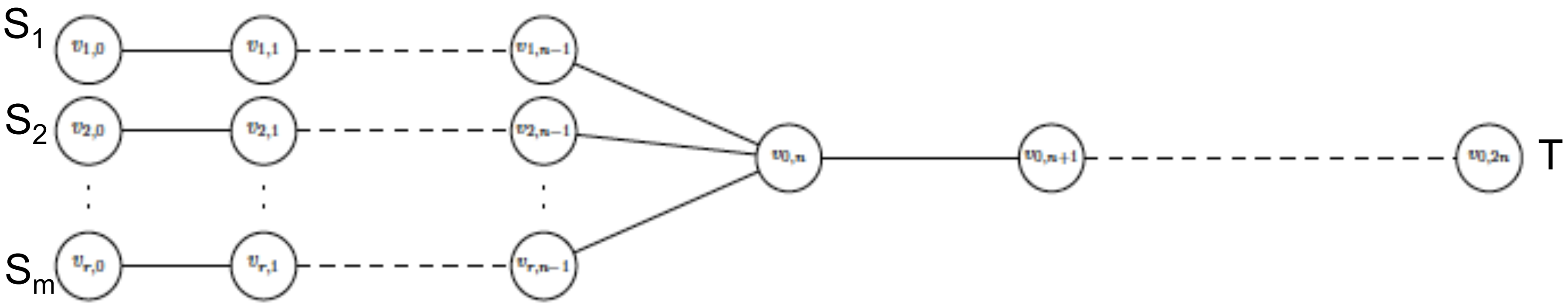
For a single message, there is an 1/ln(2) approximation algorithm.

Benefit of Collaboration

Lower Bound (BoC >= 2)

Any algorithm for delivery *without collaboration* cannot achieve an approximation ratio better than 2

For a single message, the ratio cannot de better than 1/ln(2) =1.44



BoC $\geq 1/\ln\left(\left(1+\frac{1}{2r}\right)^r\left(1+\frac{1}{2r+1}\right)\right)$

Benefit of Collaboration

Upper Bound (BoC <= 2)

There is an algorithm for delivery *without collaboration* that has approximation ratio <= 2

Proof Idea:

- Consider any optimal algorithm A and take the trajectory graph GA
- For each directed edge in GA, add a reverse edge
- In each connected component perform a Eulerian tour using the cheapest agent present in the component.
- Cost = $2.w_i$. Sum(edges in C_i) < 2 OPT(C_i)

Delivery of Multiple Packages

Delivery with Heterogenous Agents:

- **Collaboration:**

Delivery without collaboration => 2-approximation

- **Planning:**

How to plan the route of an agent delivering multiple packages?

- **Coordination:**

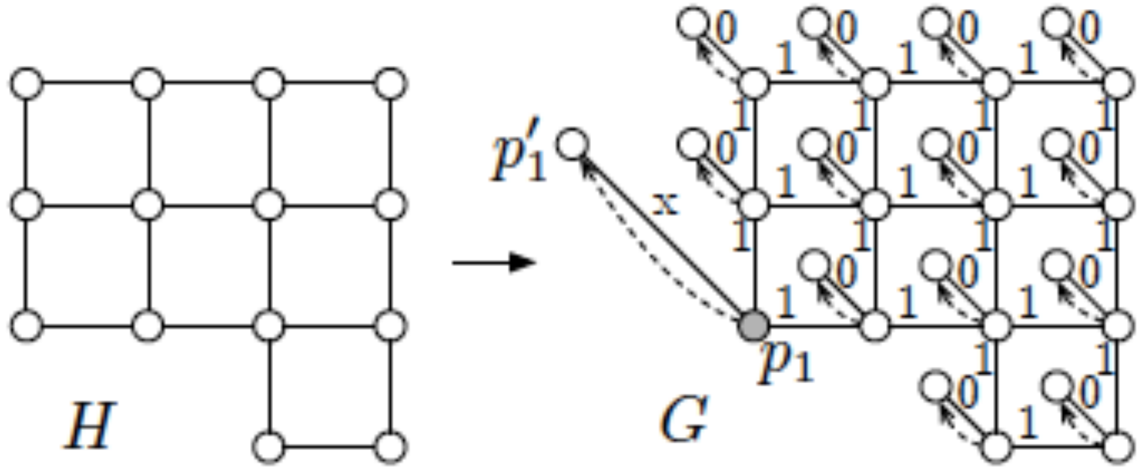
How to assign messages to agents?

Planning

Planning the schedule for a single agent

<u>Theorem</u>: Planning of delivery is NP-hard even for a single agent, and even in Planar graphs.

Reduction from Hamiltonian Cycle



Planning

Planning the schedule for a single agent

Theorem: *Planning of delivery* is NP-hard even for a single agent, and even in Planar graphs.

Inapproximability

Theorem: It is NP-hard to approximate *Planning of delivery* to within any constant less than 367/366

Planning

Approximation Algorithm for Planning Delivery

Theorem: There is a **3.5 approximation** for Planning of delivery when restricted to delivery without collaboration.

Proof Idea:

Collection -

- Each agent collects all messages assigned to it and returns home
- Using an MST of the subgraph containing the sources and p_i
- Gives a 2-approximation

Delivery -

- The agent delivers all messages using approximation of TSP (e.g. Christofide's Algorithm gives 1.5 approximation)

Delivery of Multiple Packages

Delivery with Heterogenous Agents:

- **Collaboration:**

Delivery without collaboration => 2-approximation

- **Planning:**

Planning is hard but can be approximated for each agent

- **Coordination:**

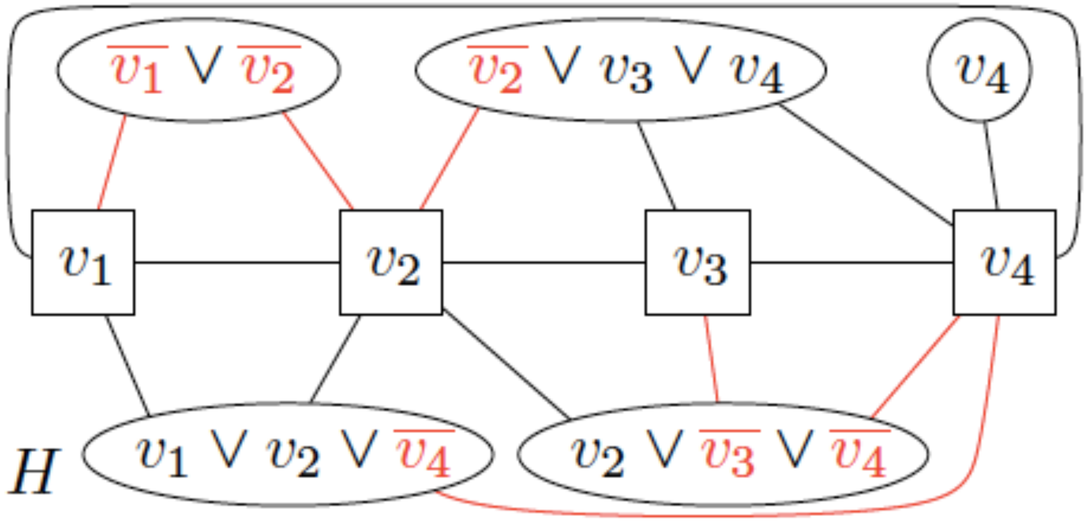
How to assign messages to agents?

Coordination

How to assign jobs to agent?

Theorem: Coordination of delivery is NP-hard even in Planar graphs.

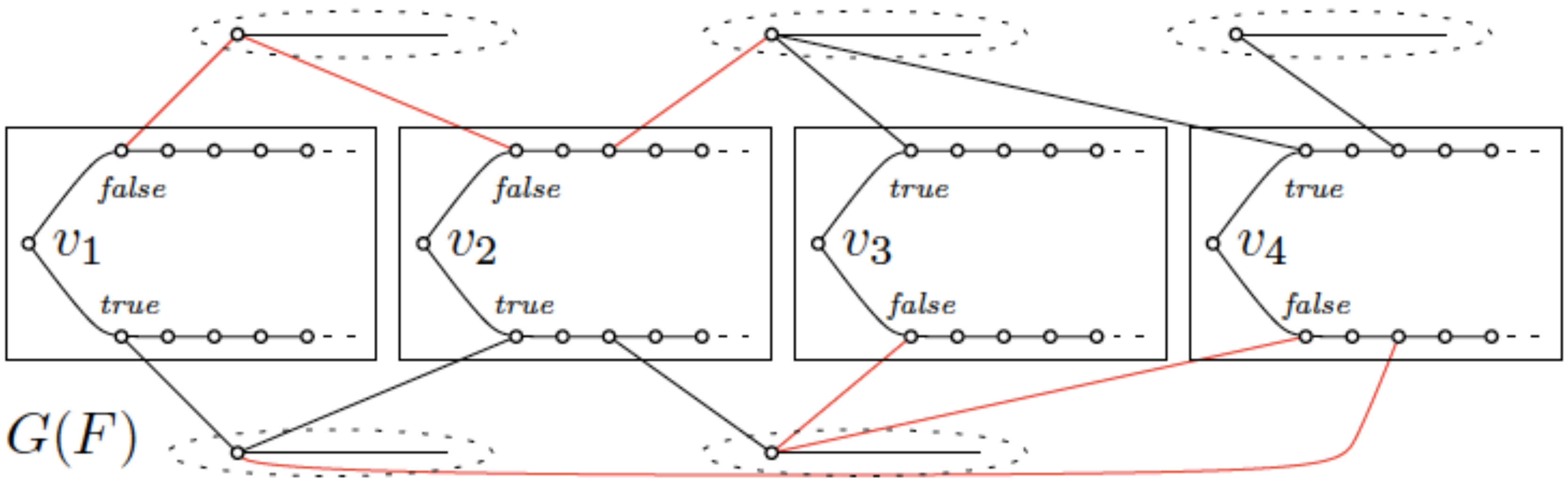
Reduction from Planar 3-SAT



Coordination

How to assign jobs to agent?

<ins>Theorem</ins>: Coordination of delivery is NP-hard even in Planar graphs.



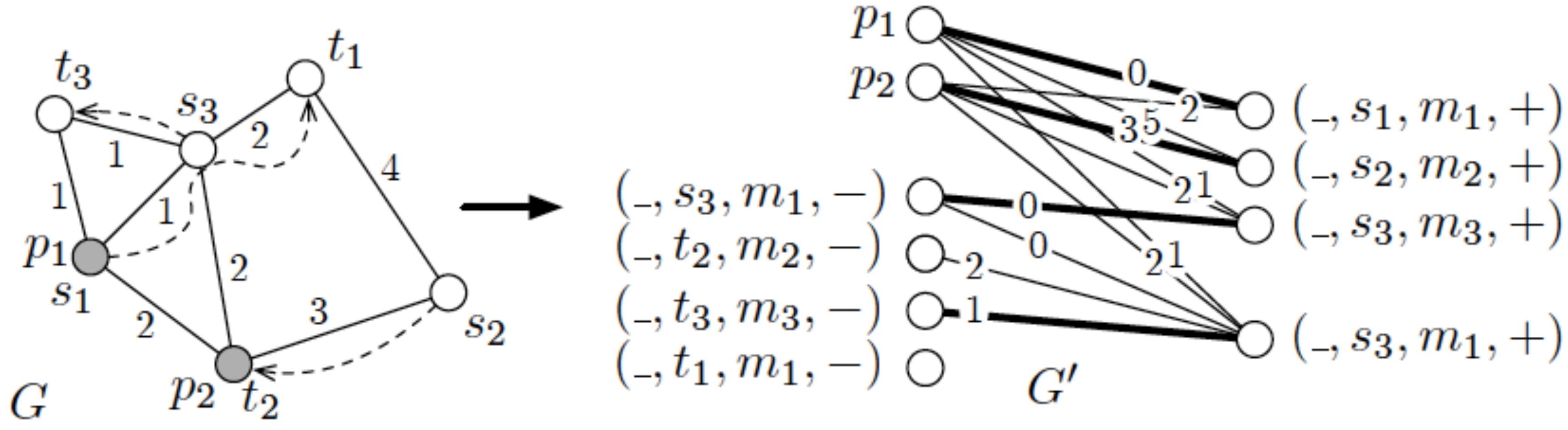
Coordination

Theorem: Coordination of delivery can be solved in polynomial time for agents with uniform weights and unit capacity.

- Each agent performs a sequence of jobs (deliveries)
- Since the weights are same, choose the closest agent for each job
- Assigning jobs to agents can be done by matching in an auxiliary graph

Coordination

Theorem: Coordination of delivery can be solved in polynomial time for agents with uniform weights and unit capacity.



Min cost matching in Bipartite graphs

Delivery of Multiple Packages

<u>Delivery with Heterogenous Agents:</u>

- **Collaboration:**

Delivery without collaboration => 2-approximation

- **Planning:**

Planning is hard but can be approximated for each agent

- **Coordination:**

NP-hard, constant approximation for uniform weight agents

Weighted Delivery

How to solve the original delivery problem with weighted agents?

Theorem: There is a polynomial time **(4 max w_i/w_j) approximation** algorithm for collaborative delivery with heterogeneous agents (with capacity 1)

- Use a restricted schedule without collaboration. (2 factor)
- Each agent uses DFS to go from one job to next. (2 factor)
- Increase the energy cost of each agent to max(w_i). (max w_i/w_j)

Related Work

Mechanism Design for Selfish Agents

- Agents can lie about their energy rate ($w_i' > w_i$)
- Objective: Design payment mechanism s.t.
  - Encourage agents to be truthful
  - Approximate (optimize) the total energy consumption
  - Payments are not too far from actual consumption

[A. Baertschi, D. Graf, P. Penna, ATMOS 2017]

Truthful Mechanisms for Delivery with Agents

Other Work on Delivery

Heterogenous Agents

- Agents have ***speeds*** and ***weights*** (Optimize Energy and Time)

Homogenous Agents : Agents can share energy

- Without constraints : One agent gathers all energy
  - Hard for general graphs, easy for trees
- With capacity constraints
  - Hard for B=2 ; Easy for chosen homebases

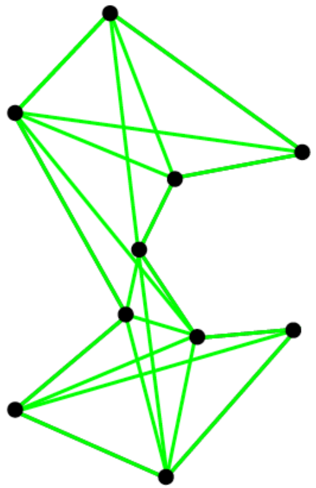
[A. Baertschi and T. Tschager; FCT 2017]

[Czyzowicz, Diks et al. Sirocco'16] ; [Bampas, D. et al. Algosensor'17]

Other Related Work

Movement Problems: Optimize TOTAL or MAX distance moved

- ***Homogeneous agents*** starting at given nodes of G
- Objective: Form specific configurations
  - ➢ Connected Subgraph
  - ➢ Clique
  - ➢ Independent Set
- General / specific graphs, Polygons



[E.Demaine, M. Hajiaghayi et al.; T. Algo 2009]

[Bilò, Gualà, Leucci, Proietti; TCS 2016]   [D. Bilo et al. Algosensors 2013]

Future Work

- ***Decentralized solutions***
  - *Selfish (non-cooperating) Agents*
  - *Online Algorithms (Exploration, …)*

- **Variants of the Model**
  - Recharging stations
  - Energy collection, energy sharing
  - Edge dependent rates, restrictions, speeds

- **Other Problems for agents with budgets**
  - Barrier Coverage, Patrolling
  - Gathering, Spreading

Thank you!