Efficient Delivery with Mobile Agents

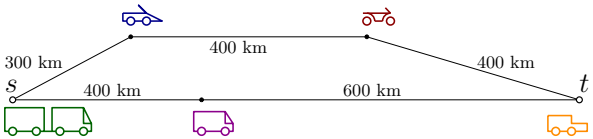
Andreas Bärtschi
NSEC/CNLS, baertschi@lanl.gov

CNLS Postdoc Seminar
April 18, 2019

Model of delivery

Setting

- undirected graph $G = (V, E)$ with edges $e \in E$ having lengths ℓ_e
- m packages: source s_i and target t_i
- k agents, each starting at node p_i, *budget* β_i, *weight* ω_i & *velocity* v_i, able to transport 1 package at a time.



Assumptions

- agents cooperate by global, centralized coordination
- handovers possible at nodes V as well as inside edges

Task: Find an *efficient delivery* in terms of

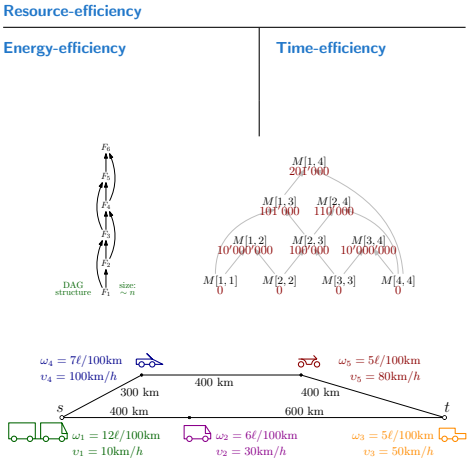
energy consumption $\mathcal{E}$ $\rightsquigarrow$ terms of form $\omega_i \cdot \ell_e$

delivery *time* $\mathcal{T}$ $\rightsquigarrow$ terms of form $\frac{1}{v_i} \cdot \ell_e$

constrained *resources* β_i $\rightsquigarrow$ range of agents

Outline

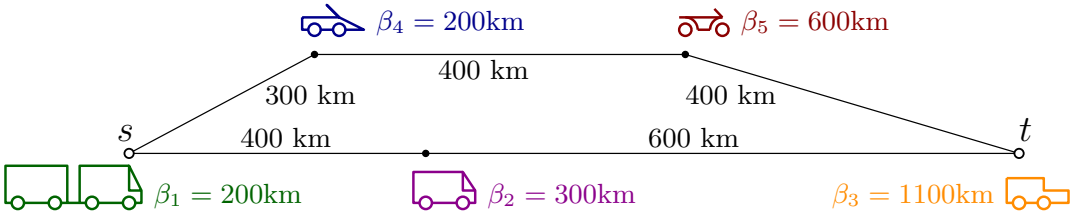
1. Examples and Results
   - Resource-efficiency: Budgets only
   - Energy-efficiency: Weights only
   - Time-efficiency: Velocities only
2. Dynamic programming
   - Technique
   - Matrix chain multiplication
3. A detailed discussion
   - Combining energy- and time-efficiency
   - OPT characterization
   - Polynomial-time algorithm



Examples and Results

Agents with budgets

The agents' resources constrain how far each agent can go (*budgets* β_i).
Can we *decide* whether there is a package delivery which respects all battery constraints?

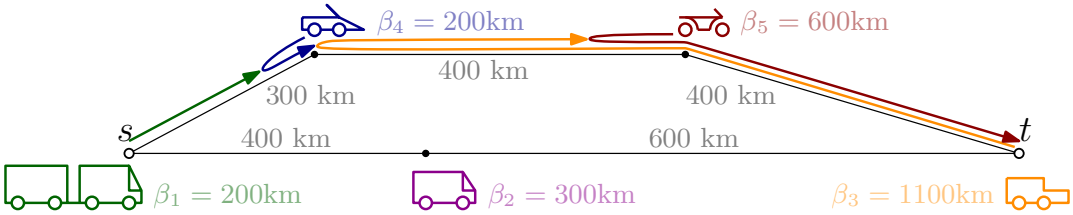


not on shortest path **in-edge-handovers** **no clear characterization**

[1] B., Chalopin, Das, Disser, Geissmann, Graf, Labourel, Mihalák: Collaborative delivery with energy-constrained mobile robots.

Agents with budgets

The agents' resources constrain how far each agent can go (*budgets* β_i).
Can we *decide* whether there is a package delivery which respects all battery constraints?



not on shortest path **in-edge-handovers** **no clear characterization**

[1] B., Chalopin, Das, Disser, Geissmann, Graf, Labourel, Mihalák: Collaborative delivery with energy-constrained mobile robots.

Resource-efficiency

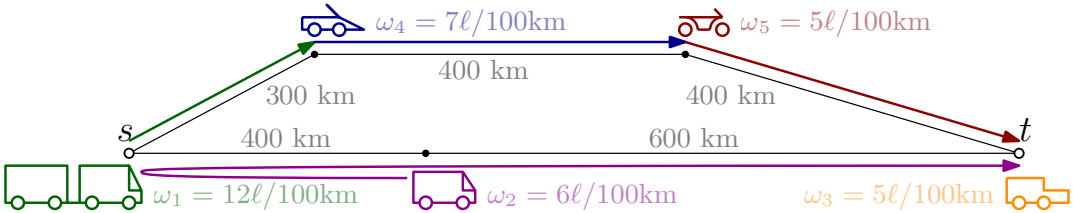
NP-hard, even for a single package ($m = 1$) on simple graphs.

Energy-efficiency

Time-efficiency

Agents with weights

Each agent has its individual energy consumption *(weight)* ω_i.
Can we *optimize* the total *energy consumption* $\mathcal{E}$ needed to deliver the package?



not on shortest path **vertex handovers** **decreasing weights**

[2] B., Chalopin, Das, Disser, Graf, Hackfeld, Penna: Energy-Efficient Delivery by Heterogeneous Mobile Agents.
[3] B., Graf, Penna: Truthful Mechanisms for Delivery with Agents.

Resource-efficiency

NP-hard, even for a single package ($m = 1$) on simple graphs.

Energy-efficiency Agents face 3 major challenges:

Collaboration: How to work together on a package?

– 2-approximation without collaboration.

Planning: Which route to take?

– NP-hard, polynomial-time 2-approximation.

Coordination: How to assign agents to packages?

– NP-hard, polynomial-time max $\frac{\omega_i}{\omega_j}$-approximation.

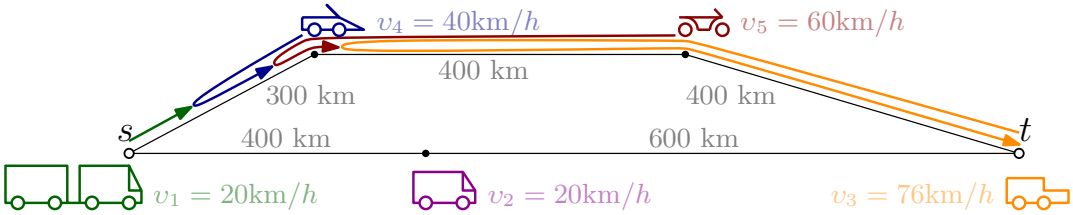
$\Rightarrow$ Polynomial-time $4 \max \frac{\omega_i}{\omega_j}$-approximation.

$\Rightarrow$ Polynomial-time algorithm for one package.

Time-efficiency

Agents with velocities

Each agent has its individual *velocity* v_i.
Can we *optimize* the total *time* $\mathcal{T}$ needed to deliver the package?



not on shortest path **(multiple) in-edge-handovers** **increasing velocities**

[4] B., Graf, Mihalák: Collective fast delivery by energy-efficient agents.

Resource-efficiency

NP-hard, even for a single package ($m = 1$) on simple graphs.

Energy-efficiency Agents face 3 major challenges:

Collaboration: How to work together on a package?

– 2-approximation without collaboration.

Planning: Which route to take?

– NP-hard, polynomial-time 2-approximation.

Coordination: How to assign agents to packages?

– NP-hard, polynomial-time $\max \frac{\omega_i}{\omega_j}$-approximation.

$\Rightarrow$ Polynomial-time $4 \max \frac{\omega_i}{\omega_j}$-approximation.

$\Rightarrow$ Polynomial-time algorithm for one package.

Time-efficiency

Existence of optima is unclear, maybe only infima exist.

$\longrightarrow$ NP-hard.

Poly-time algo. for one package.

Can we combine energy- and time-efficiency for $m = 1$?

Combining energy- and time-efficiency

Each agent has its individual *weight* ω_i and *velocity* v_i.
We want a delivery that is both efficient in its energy consumption *and* its delivery time:

- Fastest delivery among all energy-optimum ones.
  Task: lexicographically minimize $(\mathcal{E}, \mathcal{T})$.
  $\Rightarrow$ **Part 3**. [5]
- Energy-optimum delivery among all fastest ones.
  Task: lexicographically minimize $(\mathcal{T}, \mathcal{E})$.
- Tradeoff between the two measures.
  Task: minimize $\varepsilon \cdot \mathcal{T} + (1 - \varepsilon) \cdot \mathcal{E},\ \varepsilon \in (0, 1)$.

  $\Rightarrow$ NP-hard. [4]

[4] B., Graf, Mihalák: Collective fast delivery by energy-efficient agents.
[5] B., Tschager: Energy-Efficient Fast Delivery by Mobile Agents.

Dynamic programming

Dynamic programming: Technique

Programming technique which can be used if an optimal solution of a problem can be found by combining optimal solutions of subproblems: **optimal substructure**.

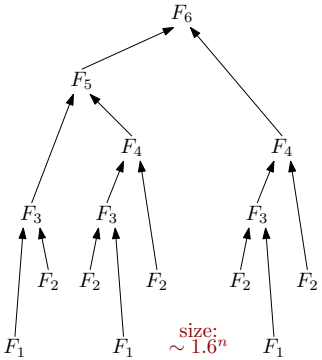
Applications:

- Graph: Shortest Paths & Dijkstra & Floyd-Warshall
- Bioinformatics: De Novo Peptide Sequencing
- Economics: Optimal Saving
- Control Theory
- Matrix chain multiplication

Toy Example: Fibonacci Sequence

- $F_n = F_{n-1} + F_{n-2}$
- $F_2 = 1$
- $F_1 = 1$

Task: Compute n-th Fibonacci number.



Dynamic programming: Technique

Programming technique which can be used if an optimal solution of a problem can be found by combining optimal solutions of subproblems: **optimal substructure**.

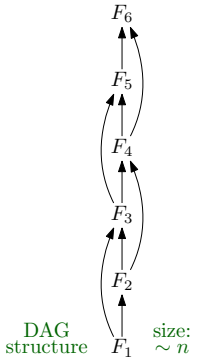
Applications:

- Graph: Shortest Paths & Dijkstra & Floyd-Warshall
- Bioinformatics: De Novo Peptide Sequencing
- Economics: Optimal Saving
- Control Theory
- Matrix chain multiplication

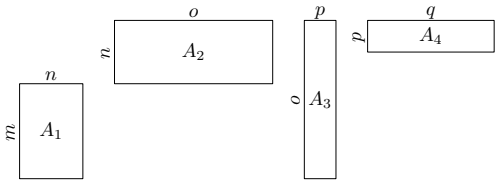
Toy Example: Fibonacci Sequence

- $F_n = F_{n-1} + F_{n-2}$
- $F_2 = 1$
- $F_1 = 1$

Task: Compute n-th Fibonacci number.



Dynamic programming: Matrix chain multiplication



$m = 100, n = 10, o = 10'000, p = 1, q = 1'000.$

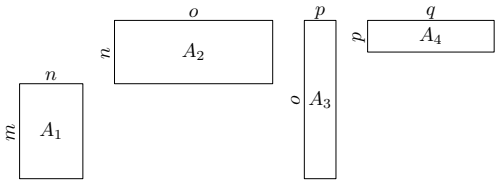
$$\overbrace{\overbrace{(A_1 \times A_2)}^{m\cdot n\cdot o=10^7} \times \overbrace{(A_3 \times A_4)}^{o\cdot p\cdot q=10^7}}^{m\cdot o\cdot q=10^9}$$

- $(A_1 \times A_2) \times (A_3 \times A_4)$ $\Rightarrow 1'020'000'000$

Find best multiplication order by:

- Testing all ways to insert parentheses: $\sim 4^{(\#\text{Matrices})}$ many!

Dynamic programming: Matrix chain multiplication



$m = 100, n = 10, o = 10'000, p = 1, q = 1'000.$ $((A_1 \times A_2) \times A_3) \times A_4$

$m \cdot p \cdot q = 10^5$

$m \cdot o \cdot p = 10^6$

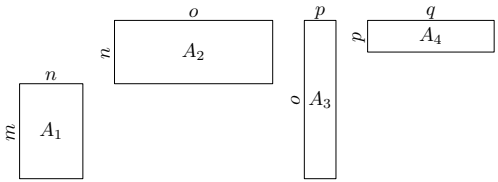
$m \cdot n \cdot o = 10^7$

- $((A_1 \times A_2) \times A_3) \times A_4$ $\Rightarrow$ $11'100'000$
- $(A_1 \times A_2) \times (A_3 \times A_4)$ $\Rightarrow 1'020'000'000$

Find best multiplication order by:

- Testing all ways to insert parentheses: $\sim 4^{(\#\text{Matrices})}$ many!

Dynamic programming: Matrix chain multiplication



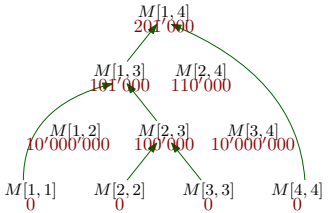
- $(A_1 \times (A_2 \times A_3)) \times A_4$
  $\Rightarrow$ $201'000$
- $((A_1 \times A_2) \times A_3) \times A_4$
  $\Rightarrow$ $11'100'000$
- $(A_1 \times A_2) \times (A_3 \times A_4)$
  $\Rightarrow 1'020'000'000$

$m = 100, n = 10, o = 10'000, p = 1, q = 1'000.$ $A_1 \times A_2 \times A_3 \times A_4$

Find best multiplication order by:

- Testing all ways to insert parentheses: $\sim 4^{(\#\text{Matrices})}$ many!
- **Dynamic Programming**: $\sim (\#\text{Matrices})^2$ subproblems:

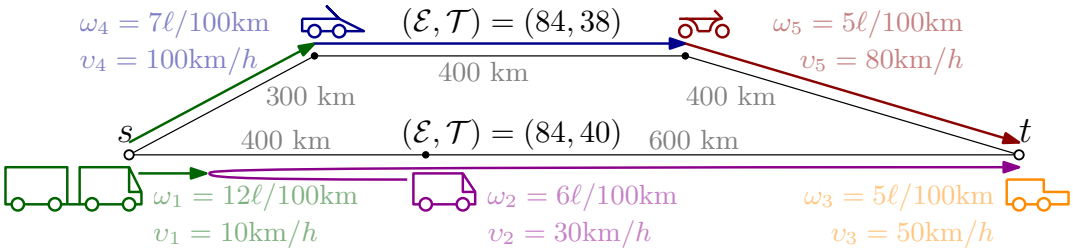
$$M[x,y] = \text{Cost of best parentheses for } A_x \times \cdots \times A_y$$
$$= \min_{x \le i < y} \left\{ \begin{matrix} M[x,i] \ + \ M[i{+}1,y] \\ + \text{ cost of multiplying two subproblems} \end{matrix} \right\}$$



A detailed discussion

Agents with weights and velocities

Each agent has its individual *weight* ω_i and *velocity* v_i.
Among all energy-optimum deliveries, can we find the fastest?



not on shortest path **0 or 1 in-edge-handovers** **decreasing tuples** (ω_i, v_i^{-1})

Characterization of an optimum delivery

Theorem (OPT characterization)

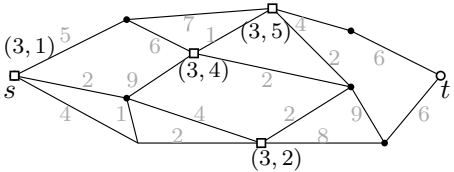
There is an optimum delivery, with the involved agents denoted by $1, \ldots, i, i+1, \ldots, k$, in which the following holds for each consecutive pair of agents:

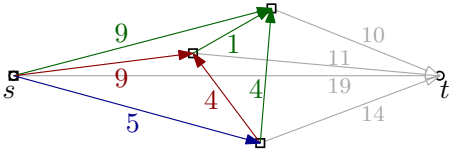
- *Decreasing weights: $\omega_i \geq \omega_{i+1}$.*
- *If $\omega_i = \omega_{i+1}$, then $v_i < v_{i+1}$.*
- *If $\omega_i = \omega_{i+1}$, then agent $i+1$ does not move without the package.*

$$\underbrace{\omega_1 = \omega_2 = \ldots}_{W_1} \underset{x}{>} \underbrace{\begin{matrix} v_i & < & v_{i+1} & < & \ldots \\ \omega_i & = & \omega_{i+1} & = & \ldots \end{matrix}}_{W_2} \underset{y}{>} \underbrace{\ldots = \omega_k}_{W_3}$$

$\Rightarrow$ First look at the problem for each weight class W_j separately.

Uniform energy consumption weights



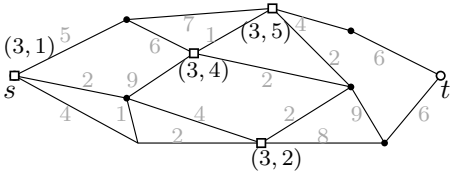


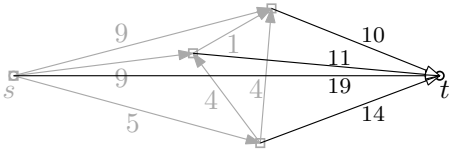
Example: 4 agents, weight 3, velocities $v_1 = 1, v_2 = 2, v_3 = 4, v_4 = 5$. Approach:

1. Move closest agent to source s. Costs $(\mathcal{E}, \mathcal{T})[p_1] = (3 \cdot 3, 3/1) = (9, 3)$.
2. Order agents by increasing velocity. Transform graph to DAG. Compute

   $(\mathcal{E}, \mathcal{T})[p_i] =$ the energy consumption $\mathcal{E}[p_i]$ and delivery time $\mathcal{T}[p_i]$ of an optimum delivery of the package from s to p_i, using only agents $1, \ldots, i-1$.

Uniform energy consumption weights





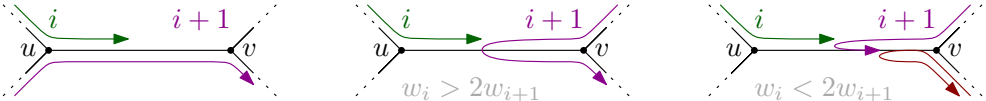
Example: 4 agents, weight 3, velocities $v_1 = 1, v_2 = 2, v_3 = 4, v_4 = 5$. Approach:

1. Move closest agent to source s. Costs $(\mathcal{E}, \mathcal{T})[p_1] = (3 \cdot 3, 3/1) = (9, 3)$.
2. Order agents by increasing velocity. Transform graph to DAG. Compute

$$(\mathcal{E}, \mathcal{T})[p_2] = (\mathcal{E}, \mathcal{T})[p_1] + (5 \cdot 3, 5/1) = (24, 8).$$
$$(\mathcal{E}, \mathcal{T})[p_3] = \min\{(\mathcal{E}, \mathcal{T})[p_1] + (9 \cdot 3, 9/1),\ (\mathcal{E}, \mathcal{T})[p_2] + (4 \cdot 3, 4/2)\} = (36, 10).$$
$$(\mathcal{E}, \mathcal{T})[p_4] = \min\{\ldots, (\mathcal{E}, \mathcal{T})[p_2] + (4 \cdot 3, 4/2), \ldots\} = (36, 10).$$

3. Compute optimum *delivery time* among energy-optimum deliveries: $(\mathcal{E}, \mathcal{T}) = (66, 12)$.

Vertex handovers



If $\omega_i \neq 2\omega_{i+1}$, then agent i does not handover the package to $i+1$ inside an edge.

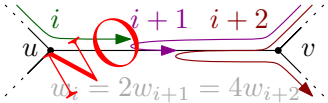
Assume we have $\omega_i \neq 2\omega_j \ \forall i, j$.

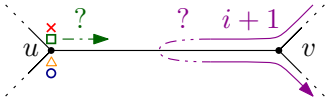
$\Rightarrow$ We can use the precomputed weight class solutions. Define the subproblems

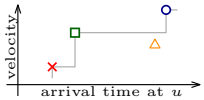
$(\mathcal{E}, \mathcal{T})[j, y] =$ the energy consumption $\mathcal{E}[j, y]$ and delivery time $\mathcal{T}[j, y]$ of an optimum delivery of the package from s up to node y, using only agents from the first j weight classes $W_1, \ldots, W_j$.

$\Rightarrow$ We can compute $(\mathcal{E}, \mathcal{T})[j, y]$ from all smaller subproblems $(\mathcal{E}, \mathcal{T})[j-1, x]$!

In-edge handovers







We can have at most one in-edge-handover per edge.
But: Which agents are involved in an in-edge-handover? Pareto frontier!

Adapt previous methods as follows:

1. Incorporate the Pareto frontier into the weight class computations.
2. Incorporate in-edge-handovers into the main dynamic program.

Running time:

- Preprocessing $poly(\mathrm{APSP} + k + |V|)$
- Per weight class: $poly(|V| \cdot k^2 + |V|^2 \cdot k)$
- Main dynamic program: $poly(k \cdot |V| \cdot |V|)$

$\Big\}$ $poly(k + |V|^3)$

Resource-efficiency, $m = 1$ NP-hard on simple graphs.

Energy-efficiency	**Time-efficiency**
Collaboration:	Existence of optima is unclear (maybe only infima)
2-apx without collaboration	
Planning:	
NP-hard, poly-time 2-apx	NP-hard
Coordination:	
NP-hard, poly-time max $\frac{\omega_i}{\omega_j}$-apx	
Poly-time algorithm for $m = 1$	Poly-time algo. for $m = 1$

Combining energy- and time-efficiency, $m = 1$

Some versions NP-hard, but lexicographically minimizing $(\mathcal{E}, \mathcal{T})$ can be done in polynomial time.

https://xkcd.com/1925/