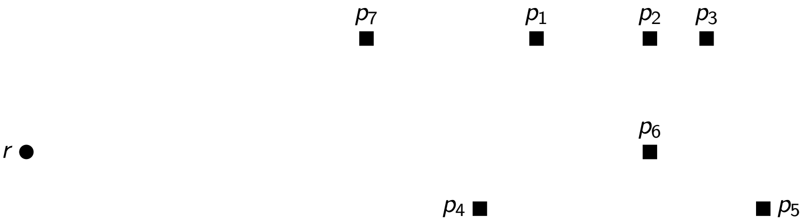
Vehicle Routing with Subtours

Stephan Held, Jochen Könemann, and Jens Vygen

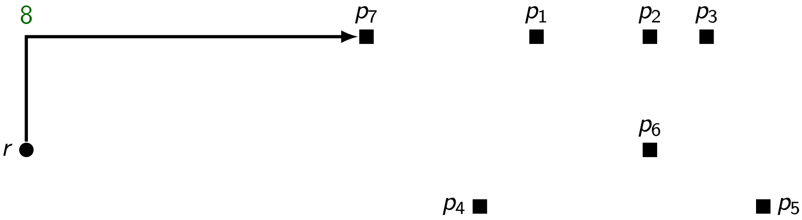
Banff, September 24, 2018

Parcel delivery with a deadline



- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.

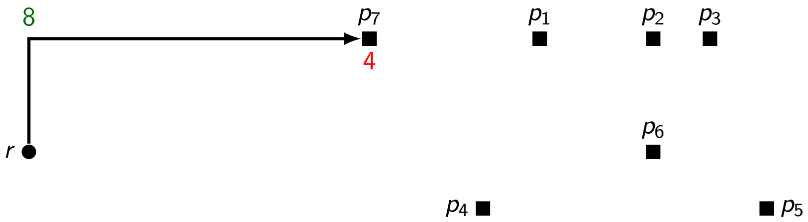
Parcel delivery with a deadline



latest delivery time $= 8$

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance

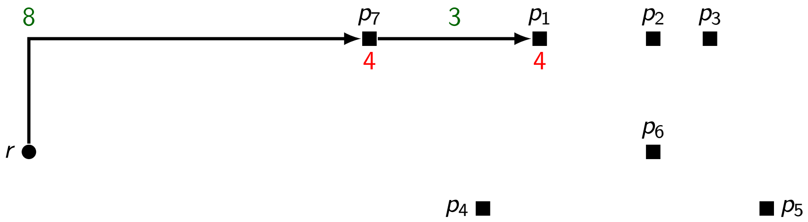
Parcel delivery with a deadline



latest delivery time $= 12 = 8 + 1 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

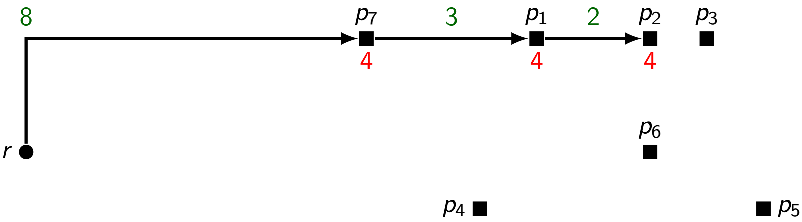
Parcel delivery with a deadline



latest delivery time $= 19 = 11 + 2 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

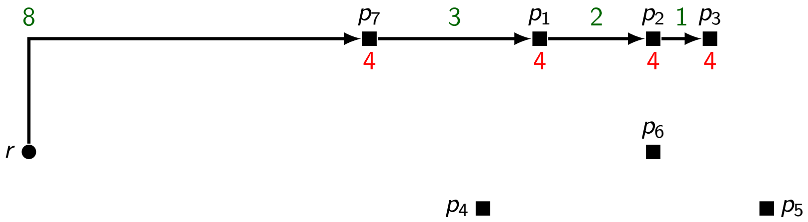
Parcel delivery with a deadline



latest delivery time $= 25 = 13 + 3 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

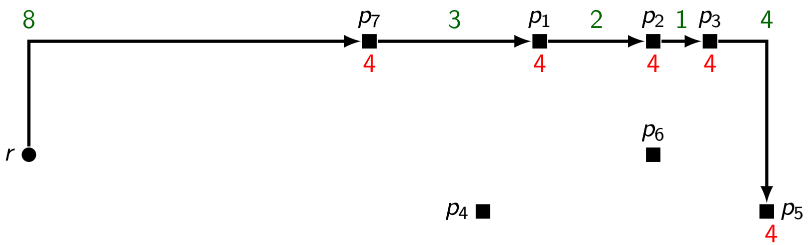
Parcel delivery with a deadline



latest delivery time $= 30 = 14 + 4 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

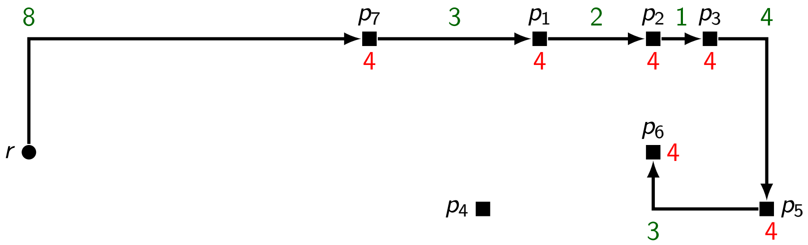
Parcel delivery with a deadline



latest delivery time $= 38 = 18 + 5 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

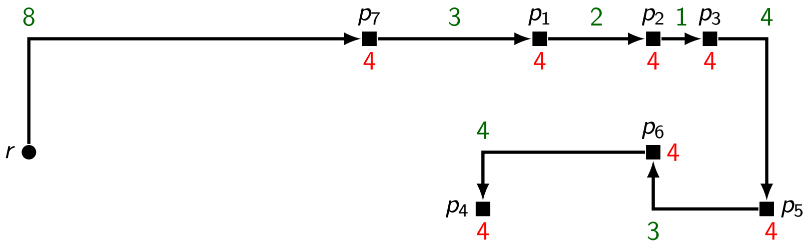
Parcel delivery with a deadline



latest delivery time $= 45 = 21 + 6 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

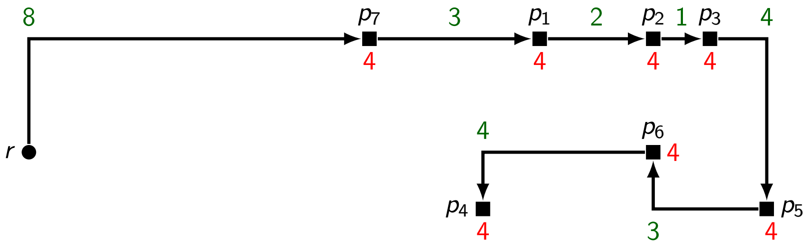
Parcel delivery with a deadline



latest delivery time $= 53 = 25 + 7 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

Parcel delivery with a deadline

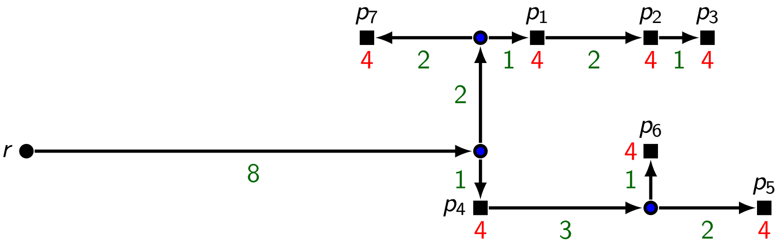


latest delivery time $= 53 = 25 + 7 \cdot 4 =$ drive time+delivery time

- A root r, a finite set P of parcels, $n := |P|$.
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$.
- unit drive time per distance
- $\delta =$ delivery time per parcel

Can we reduce the latest delivery time?

Faster delivery with subtours



latest delivery time $= 26 = 14 + 3 \cdot 4$ (attained at p_3).

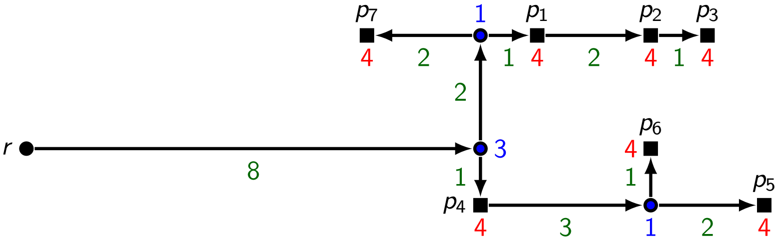
- A root r, a finite set P of parcels, a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$,
- unit drive time per distance $c(v, w)$
- $\delta =$ delivery time per parcel,

Faster delivery with subtours

latest delivery time $= 26 = 14 + 3 \cdot 4$ (attained at p_3).

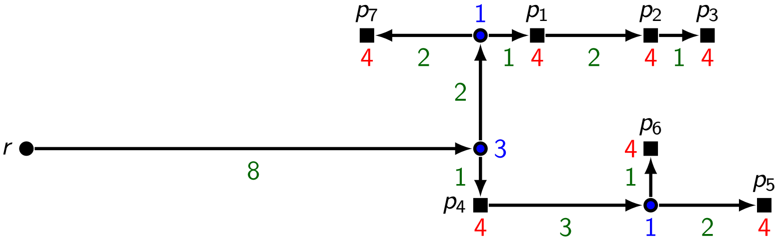
- A root r, a finite set P of parcels, a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$,
- unit drive time per distance $c(v, w)$
- $\delta =$ delivery time per parcel,

Faster delivery with subtours



- A root r, a finite set P of parcels,
  a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$,
- unit drive time per distance $c(v, w)$
- $\delta =$ delivery time per parcel,
- unit handover delay per parcel that is moved between vehicles,

Faster delivery with subtours

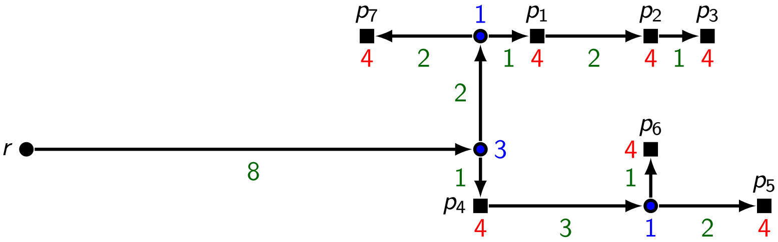


latest delivery time $= 30 = 14 + 3 \cdot 4 + 4$ (attained at p_3).

- A root r, a finite set P of parcels, a metric space (M, c), and a map $\mu : \{r\} \cup P \to M$,
- unit drive time per distance $c(v, w)$
- $\delta =$ delivery time per parcel,
- unit handover delay per parcel that is moved between vehicles,

Assumption $\delta \geq 1$ (i.e. delivery time $\geq$ handover time).

Faster delivery with subtours



A schedule (A, W, μ) is given by an arborescence (W, A) rooted at r with $P \subset W$ and an extension $\mu : W \setminus (P \cup \{r\}) \to M$.

Degree constraints for well-defined delays:

- $|\delta^+(w)| \leq 2$ $(w \in W)$,
- $|\delta^+(p)| \leq 1$ $(p \in P)$.

Multiple vertices are allowed at one position.

Delay and cost of a schedule (W, A, μ)

$$\begin{aligned}
\mathrm{delay}_{(W,A,\mu)}(r,y) :=\; & c(A_{ry},\mu) + \delta|P \cap W_{ry}| \\
& + \sum_{w \in W_{ry} \cap W_2} \min_{(w,x)\in\delta^+(w)} |P \cap W_{x*}| \qquad (y \in W) \\
\mathrm{delay}(W,A,\mu) :=\; & \max_{p \in P} \mathrm{delay}_{(W,A,\mu)}(r,p)
\end{aligned}$$

$$\begin{aligned}
\mathrm{cost}(W,A,\mu) :=\; & \sum_{(x,y)\in A} c(\mu(x),\mu(y)) + \sigma \cdot \#\text{ leaves in } (W,A) \\
=\; & \text{travel cost} + \text{setup cost},
\end{aligned}$$

constant setup cost per vehicle $\sigma \geq 0$,

$$\begin{aligned}
(W_{x*}, A_{x*}) &= \text{sub-arborescence rooted at } x \in W, \\
(W_{xy}, A_{xy}) &= \text{the } x\text{-}y \text{ sub-path for } x, y \in W, \\
W_i &= \text{set of vertices with out-degree } i \in \{0,1,2\}.
\end{aligned}$$

Main Result

Theorem (H., Könemann, and Vygen)

Given a deadline $\Delta > 0$ and a feasible instance, we can compute a schedule with delay at most $(1+\epsilon)\Delta$ and cost $\mathcal{O}(1+\frac{1}{\epsilon})\mathrm{OPT}$ in polynomial time, where OPT is the minimum cost of a schedule with latest delivery $\leq \Delta$.

Related Work

- shallow-light trees ($\sigma = 0$, $c >> \delta$)
  (Awerbuch, Baratz & Peleg '90, Cong et al. '92, Khuller, Raghavachari & Young '95, H. & Rotter '13)
- bounded-latency problem ($c >> \delta, \sigma >> c$)
  (Jothi and Raghavachari '07)
- distance-constrained vehicle routing problem
  (Khuller, Malekian & Mestre '11, Nagarajan & Ravi '12, Friggstad & Swamy '14)
- Min-Max tree/path/tour cover
  (Even, Garg, Könemann, Ravi, Sinha '04, Arkin, Hassin & Levin '06, Xu, Xu& Li '10, Khani & Salavatipour '14).

Typical strategy: 1) Compute cheap global solution, 2) split into sub-solutions at delay violations, 3) combine sub-solutions.

Related Work

- shallow-light trees ($\sigma = 0$, $c >> \delta$)
  (Awerbuch, Baratz & Peleg '90, Cong et al. '92, Khuller, Raghavachari & Young '95, H. & Rotter '13)
- bounded-latency problem ($c >> \delta, \sigma >> c$)
  (Jothi and Raghavachari '07)
- distance-constrained vehicle routing problem
  (Khuller, Malekian & Mestre '11, Nagarajan & Ravi '12, Friggstad & Swamy '14)
- Min-Max tree/path/tour cover
  (Even, Garg, Könemann, Ravi, Sinha '04, Arkin, Hassin & Levin '06, Xu, Xu& Li '10, Khani & Salavatipour '14).

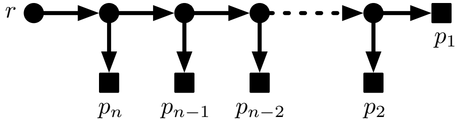
Typical strategy: 1) Compute cheap global solution, 2) split into sub-solutions at delay violations, 3) combine sub-solutions.

Here, naïve adaption of strategy fails due to the handover delays.

Checking feasibility: caterpillar schedules

Theorem (H., Könemann, Vygen)

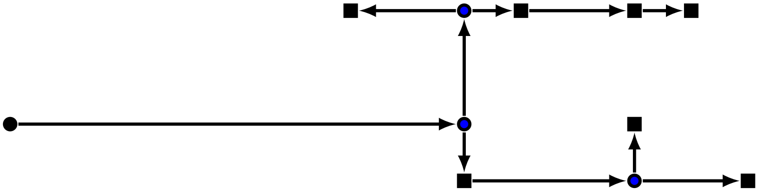
There exists a schedule (W, A, μ) with minimum delay such that (W, A) is a caterpillar and $\mu(w) = \mu(r)$ for all $w \in W_2$.



caterpillar: deliveries occur at leaves and the subgraph induced by the vertices with out-degree 2 is a path.

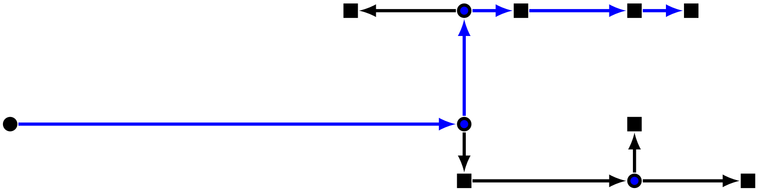
Proof for: ∃ caterpillar schedule with minimum delay

- Take any schedule



Proof for: ∃ caterpillar schedule with minimum delay

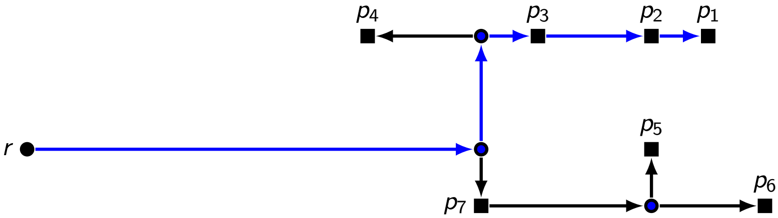
- Take any schedule



- $P =$ path following the majority of the parcels (initial vehicle)

Proof for: $\exists$ caterpillar schedule with minimum delay

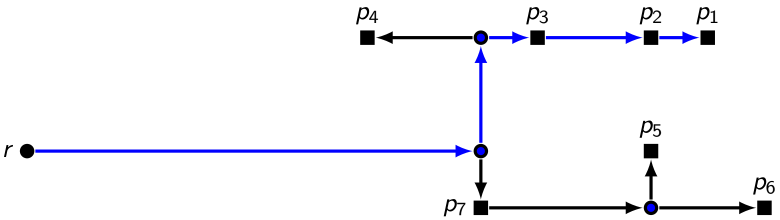
- Take any schedule



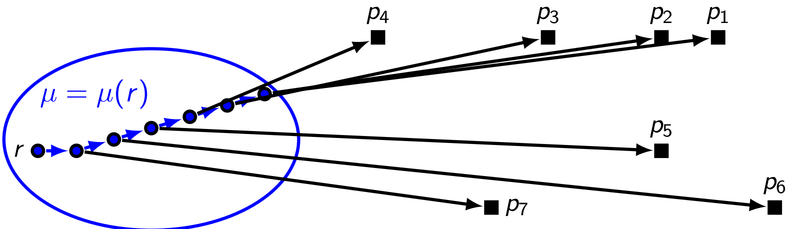
- $P =$ path following the majority of the parcels (initial vehicle)
- $p_1, \ldots, p_n$ reversely ordered as leaving the initial vehicle.

Proof for: ∃ caterpillar schedule with minimum delay

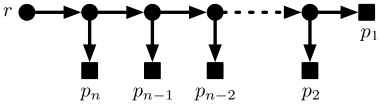
- Take any schedule



- P = path following the majority of the parcels (initial vehicle)
- $p_1, \ldots, p_n$ reversely ordered as leaving the initial vehicle.
- Caterpillar with all internal vertices located at $\mu(r)$ has no more delay ($\delta \geq 1$!).



Checking feasibility: consequences



Corollary

We can decide feasibility in time $\mathcal{O}(n \log n + \theta n)$*, where* θ *is the time to evaluate distances in* (M, c)*.*

Corollary

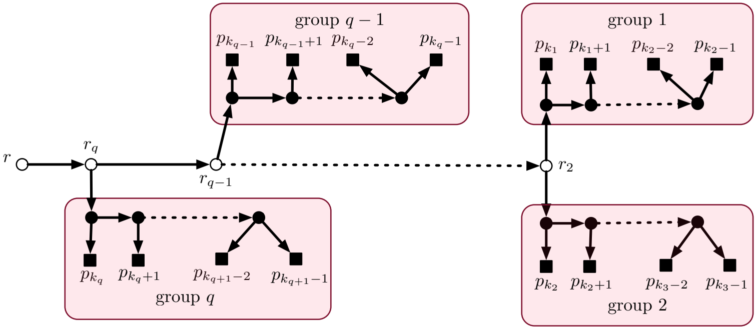
For any feasible instance we have

$$\Delta \geq \delta + \min\{|Q|, n-1\} + \min_{q \in Q} c(r, q)$$

for every nonempty subset $Q \subseteq P$*.*

Bicriteria approximation

1. Grouping into groups with similar distance from r.
2. Bottom-level caterpillar for each group.
3. Top-level caterpillar connecting the groups.
4. Subtour merging within groups to reduce $\#vehicles$.



Bicriteria approximation: grouping

1. Compte a minimum spanning tree for $\{r\} \cup P$.
2.
   - $s := \arg\max\{c(r,p) : p \in P\}$,
     double edges except those on the r-s-path,
   - $(\{r\} \cup P, F_0) :=$ Eulerian r-s-walk.
     We will choose $F \subseteq F_0$. $\Rightarrow c(F) \leq c(F_0) \leq 2\mathrm{MST} - c(r,s)$.
3. Remove r and the first edge.
   Split the remaining r-s-path into maximal paths s.t.
   - path length $\leq \epsilon\Delta$ and
   - $\#$ parcels on path $\leq 1 + \epsilon\Delta$.

   forest of paths $\rightsquigarrow (P, F)$

The length bound can be exceeded at most $\frac{c(F_0)}{\epsilon\Delta}$ times.
The parcel bound can be exceeded at most $\frac{n}{\epsilon\Delta}$ times.
Observation:

$$q := \#\text{ paths} \leq 1 + \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}.$$

Bicriteria approximation: grouping

Groups are defined as the maximal paths of (P, F).

Corollary: $c(r, p) \leq c(r, p') + \epsilon\Delta$ for p, p' in the same group.

Bicriteria approximation: grouping

Groups are defined as the maximal paths of (P, F).

Corollary: $c(r,p) \leq c(r,p') + \epsilon\Delta$ for p, p' in the same group.

$d(p) := \max\{c(r,p') : p \text{ and } p' \text{ are in the same group}\}$.

Order groups/parcels $P = \{p_1, \ldots, p_n\}$ s.t.

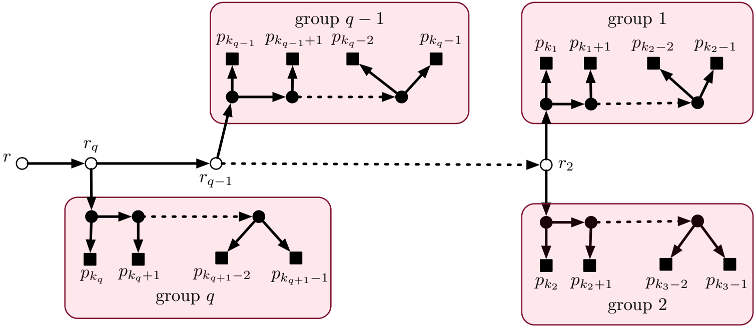
- $d(p_1) \leq \cdots \leq d(p_n)$ and
- $F \subseteq \{\{p_i, p_{i+1}\} : i = 1, \ldots, n-1\}$,

($\Rightarrow$ groups are consecutive subsequences)

Bicriteria approximation: a first schedule S_1

Let $k_1 =< k_2 < \cdots < k_{q+1} = n + 1$ s.t.
$\{p_{k_i}, \ldots, p_{k_i+1} - 1\}$ are the groups $(1 \leq i \leq q)$.

Schedule S_1:



Top-level bifurcation nodes $r_2, \ldots, r_q$ are placed at $\mu(r)$.
Bottom-level bifurcation nodes are placed at the splitted parcel.

Bicriteria approximation: delay of S_1

Lemma: If the instance is feasible, then $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.

Proof: The maximum delay in group $i \in \{1, \ldots, q\}$ is at most

$$\left(c(r, p_{k_i}) + \sum_{l=k_i}^{k_{i+1}-2} c(p_l, p_{l+1}) \right) + \left(n - k_{\max\{2,i\}} + 1\right) + \left(k_{i+1} - k_i - 1\right) + \delta$$

$$\leq \left(d(p_{k_i}) + \epsilon\Delta\right) + \left(n - k_{\max\{2,i\}} + 1\right) + \epsilon\Delta + \delta. \qquad (*)$$

$\exists j \in \{k_i, \ldots, n\}$ with

$$\begin{aligned} \Delta &\geq c(r, p_j) + n - \max\{2, k_i\} + 1 + \delta \\ &\geq d(p_{k_i}) - \epsilon\Delta + n - k_{\max\{2,i\}} + 1 + \delta. \end{aligned} \qquad (**)$$

$(*) + (**)$ prove: $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.

Bicriteria approximation: delay of S_1

Lemma: If the instance is feasible, then $\mathrm{delay}(S_1) \le (1+3\epsilon)\Delta$.
Proof: The maximum delay in group $i \in \{1,\dots,q\}$ is at most

$$\left(c(r,p_{k_i}) + \sum_{l=k_i}^{k_{i+1}-2} c(p_l,p_{l+1})\right) + \big(n - k_{\max\{2,i\}} + 1\big) + \big(k_{i+1} - k_i - 1\big) + \delta$$

$$\le \big(d(p_{k_i}) + \epsilon\Delta\big) + \big(n - k_{\max\{2,i\}} + 1\big) + \epsilon\Delta + \delta. \qquad (*)$$

$\exists j \in \{k_i,\dots,n\}$ with

$$\begin{aligned} \Delta &\ge c(r,p_j) + n - \max\{2,k_i\} + 1 + \delta \\ &\ge d(p_{k_i}) - \epsilon\Delta + n - k_{\max\{2,i\}} + 1 + \delta. \end{aligned} \qquad (**)$$

$(*) + (**)$ prove: $\mathrm{delay}(S_1) \le (1+3\epsilon)\Delta$.

Bicriteria approximation: delay of S_1

Lemma: If the instance is feasible, then $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.
Proof: The maximum delay in group $i \in \{1, \dots, q\}$ is at most

$$\left(c(r, p_{k_i}) + \sum_{l=k_i}^{k_{i+1}-2} c(p_l, p_{l+1}) \right) + \left(n - k_{\max\{2,i\}} + 1\right) + \left(k_{i+1} - k_i - 1\right) + \delta$$
$$\leq \left(d(p_{k_i}) + \epsilon\Delta\right) + \left(n - k_{\max\{2,i\}} + 1\right) + \epsilon\Delta + \delta. \qquad (*)$$

$\exists j \in \{k_i, \dots, n\}$ with

$$\begin{aligned} \Delta &\geq c(r, p_j) + n - \max\{2, k_i\} + 1 + \delta \\ &\geq d(p_{k_i}) - \epsilon\Delta + n - k_{\max\{2,i\}} + 1 + \delta. \end{aligned} \qquad (**)$$

$(*) + (**)$ prove: $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.

Bicriteria approximation: delay of S_1

Lemma: If the instance is feasible, then $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.
Proof: The maximum delay in group $i \in \{1,\ldots,q\}$ is at most

$$\left(c(r,p_{k_i}) + \sum_{l=k_i}^{k_{i+1}-2} c(p_l,p_{l+1})\right) + \left(n - k_{\max\{2,i\}} + 1\right) + \left(k_{i+1} - k_i - 1\right) + \delta$$
$$\leq \left(d(p_{k_i}) + \epsilon\Delta\right) + \left(n - k_{\max\{2,i\}} + 1\right) + \epsilon\Delta + \delta. \qquad (*)$$

$\exists j \in \{k_i,\ldots,n\}$ with

$$\begin{aligned} \Delta &\geq c(r,p_j) + n - \max\{2,k_i\} + 1 + \delta \\ &\geq d(p_{k_i}) - \epsilon\Delta + n - k_{\max\{2,i\}} + 1 + \delta. \end{aligned} \qquad (**)$$

$(*) + (**)$ prove: $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.

Bicriteria approximation: delay of S_1

Lemma: If the instance is feasible, then $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.
Proof: The maximum delay in group $i \in \{1, \ldots, q\}$ is at most

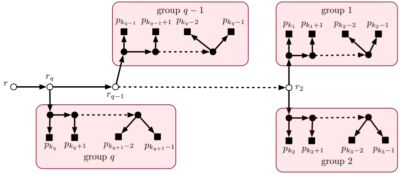
$$\left(c(r, p_{k_i}) + \sum_{l=k_i}^{k_{i+1}-2} c(p_l, p_{l+1}) \right) + \left(n - k_{\max\{2,i\}} + 1\right) + \left(k_{i+1} - k_i - 1\right) + \delta$$
$$\leq \left(d(p_{k_i}) + \epsilon\Delta\right) + \left(n - k_{\max\{2,i\}} + 1\right) + \epsilon\Delta + \delta. \qquad (*)$$

$\exists j \in \{k_i, \ldots, n\}$ with

$$\begin{aligned} \Delta &\geq c(r, p_j) + n - \max\{2, k_i\} + 1 + \delta \\ &\geq d(p_{k_i}) - \epsilon\Delta + n - k_{\max\{2,i\}} + 1 + \delta. \end{aligned} \qquad (**)$$

$(*) + (**)$ prove: $\mathrm{delay}(S_1) \leq (1+3\epsilon)\Delta$.

Bicriteria approximation: length of S_1



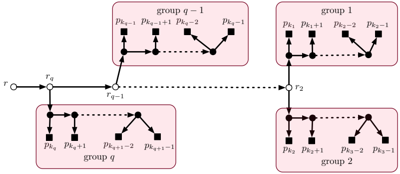
Lemma: S_1 has length at most $(2 + \frac{2}{\epsilon})\mathrm{MST}$.

Proof: The length is at most

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{q} c(r, p_{k_i}) + c(F) \\
\leq \;& qc(r,s) + 2\mathrm{MST} - c(r,s) \\
\leq \;& \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta} c(r,s) + 2\,\mathrm{MST} \\
\leq \;& \frac{\Delta}{\epsilon\Delta} c(r,s) + \frac{2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\Delta + 2\,\mathrm{MST} \\
= \;& \left(2 + \frac{2}{\epsilon}\right) \mathrm{MST}.
\end{aligned}
$$

□

Bicriteria approximation: length of S_1

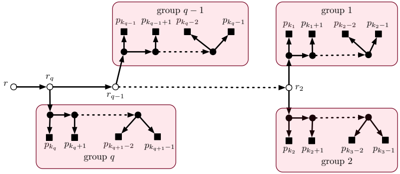


Lemma: S_1 has length at most $(2+\frac{2}{\epsilon})\mathrm{MST}$.

Proof: The length is at most

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{q} c(r, p_{k_i}) + c(F) \\
\leq \quad & qc(r,s) + 2\mathrm{MST} - c(r,s) \\
\leq \quad & \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\, c(r,s) + 2\,\mathrm{MST} \\
\leq \quad & \frac{\Delta}{\epsilon\Delta} c(r,s) + \frac{2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\Delta + 2\,\mathrm{MST} \\
= \quad & \left(2 + \frac{2}{\epsilon}\right)\,\mathrm{MST}.
\end{aligned}
$$

Bicriteria approximation: length of S_1

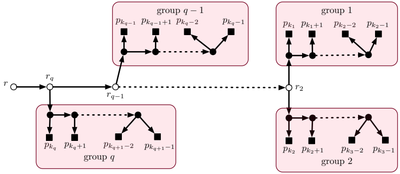


Lemma: S_1 has length at most $(2 + \frac{2}{\epsilon})\mathrm{MST}$.
Proof: The length is at most

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{q} c(r, p_{k_i}) + c(F) \\
\leq \quad & qc(r,s) + 2\mathrm{MST} - c(r,s) \\
\leq \quad & \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta} c(r,s) + 2\,\mathrm{MST} \\
\leq \quad & \frac{\Delta}{\epsilon\Delta} c(r,s) + \frac{2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\Delta + 2\,\mathrm{MST} \\
= \quad & \left(2 + \frac{2}{\epsilon}\right)\,\mathrm{MST}.
\end{aligned}
$$

□

Bicriteria approximation: length of S_1

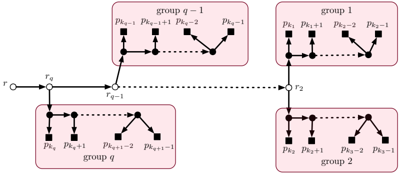


Lemma: S_1 has length at most $(2 + \frac{2}{\epsilon})\mathrm{MST}$.
Proof: The length is at most

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{q} c(r, p_{k_i}) + c(F) \\
\leq \quad & qc(r,s) + 2\mathrm{MST} - c(r,s) \\
\leq \quad & \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\, c(r,s) + 2\,\mathrm{MST} \\
\leq \quad & \frac{\Delta}{\epsilon\Delta} c(r,s) + \frac{2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\Delta + 2\,\mathrm{MST} \\
= \quad & \left(2 + \frac{2}{\epsilon}\right)\,\mathrm{MST}.
\end{aligned}
$$

□

Bicriteria approximation: length of S_1

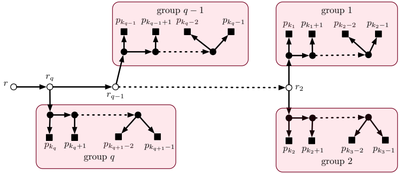


Lemma: S_1 has length at most $(2+\frac{2}{\epsilon})\mathrm{MST}$.

Proof: The length is at most

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{q} c(r, p_{k_i}) + c(F) \\
\leq\ & qc(r,s) + 2\mathrm{MST} - c(r,s) \\
\leq\ & \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\, c(r,s) + 2\,\mathrm{MST} \\
\leq\ & \frac{\Delta}{\epsilon\Delta} c(r,s) + \frac{2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\Delta + 2\,\mathrm{MST} \\
=\ & \left(2 + \frac{2}{\epsilon}\right)\mathrm{MST}.
\end{aligned}
$$

Bicriteria approximation: length of S_1



Lemma: S_1 has length at most $(2 + \frac{2}{\epsilon})\mathrm{MST}$.

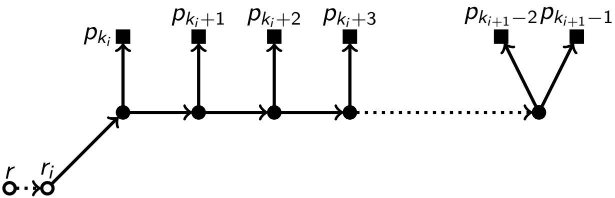
Proof: The length is at most

$$
\begin{aligned}
& \textstyle\sum_{i=1}^{q} c(r, p_{k_i}) + c(F) \\
\leq \quad & qc(r,s) + 2\mathrm{MST} - c(r,s) \\
\leq \quad & \frac{n + 2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\, c(r,s) + 2\,\mathrm{MST} \\
\leq \quad & \frac{\Delta}{\epsilon\Delta} c(r,s) + \frac{2\,\mathrm{MST} - c(r,s)}{\epsilon\Delta}\Delta + 2\,\mathrm{MST} \\
= \quad & \left(2 + \frac{2}{\epsilon}\right)\,\mathrm{MST}.
\end{aligned}
$$

□

Saving vehicles

S_1 is "short" and "fast", but uses one vehicle per parcel.



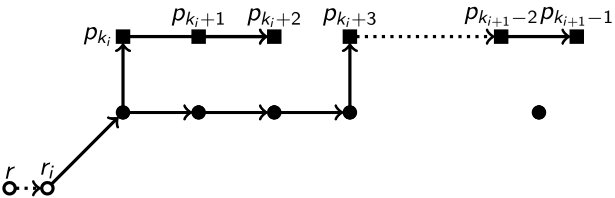
Serve up to $m := 1 + \lfloor \frac{\epsilon\Delta}{\delta} \rfloor$ parcels within a group by one vehicle.
$\rightsquigarrow S_2$.

Lemma

S_2 has delay at most $(1 + 4\epsilon)\Delta$ and length at most $(4 + \frac{2}{\epsilon})\mathrm{MST}$. It has at most $1 + \frac{2}{\epsilon}\left(\frac{\mathrm{MST}+n\delta}{\Delta}\right)$ vehicles.

Saving vehicles

S_1 is "short" and "fast", but uses one vehicle per parcel.



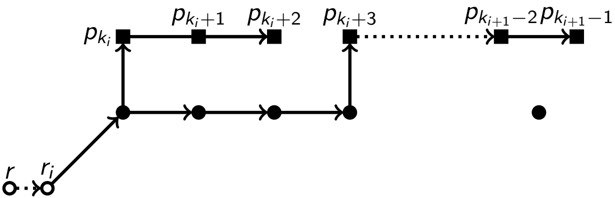
Serve up to $m := 1 + \lfloor \frac{\epsilon\Delta}{\delta} \rfloor$ parcels within a group by one vehicle.
$\rightsquigarrow S_2$.

Lemma

S_2 has delay at most $(1+4\epsilon)\Delta$ and length at most $(4+\frac{2}{\epsilon})\mathrm{MST}$. It has at most $1+\frac{2}{\epsilon}\left(\frac{\mathrm{MST}+n\delta}{\Delta}\right)$ vehicles.

Saving vehicles

S_1 is “short” and ”fast”, but uses one vehicle per parcel.



Serve up to $m := 1 + \lfloor \frac{\epsilon\Delta}{\delta} \rfloor$ parcels within a group by one vehicle.
$\rightsquigarrow S_2$.

Lemma

S_2 has delay at most $(1 + 4\epsilon)\Delta$ and length at most $(4 + \frac{2}{\epsilon})\mathrm{MST}$. It has at most $1 + \frac{2}{\epsilon}\left(\frac{\mathrm{MST}+n\delta}{\Delta}\right)$ vehicles.

A lower bound

Lemma

Every feasible schedule has

- *length at least* $\frac{1}{2}\mathrm{MST}$ *and*
- *uses at least* $\frac{\frac{1}{2}\mathrm{MST}+n\delta}{\Delta}$ *vehicles.*

Proof.

Steiner ratio $\Rightarrow$ length bound.
Let (W^*, A^*, μ^*) be a schedule with l^* vehicles numbered $1, \dots, l^*$.
$D_i :=$ delay of the last parcel delivered by vehicle i. $\Rightarrow D_i \leq \Delta$.

$$\frac{1}{2}\mathrm{MST} + n\delta \;\leq\; c(A^*, \mu^*) + n\delta \;\leq\; \sum_{i=1}^{l^*} D_i \;\leq\; l^*\Delta.$$

□

Combining upper and lower bound

Lemma (Upper bound)

S_2 has delay at most $(1 + 4\epsilon)\Delta$ and length at most $(4 + \frac{2}{\epsilon})\mathrm{MST}$. It has at most $1 + \frac{2}{\epsilon}\left(\frac{\mathrm{MST}+n\delta}{\Delta}\right)$ vehicles.

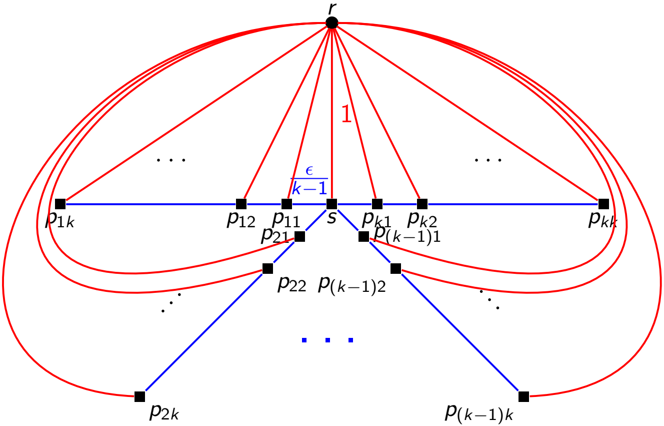
Lemma (Lower bound)

Every feasible schedule has length at least $\frac{1}{2}\mathrm{MST}$ and uses at least $\frac{\frac{1}{2}\mathrm{MST}+n\delta}{\Delta}$ vehicles.

Theorem (H., Könemann, Vygen)

Given a deadline $\Delta > 0$ and a feasible instance, we can compute a schedule with delay at most $(1 + \epsilon)\Delta$ and cost $\mathcal{O}(1 + \frac{1}{\epsilon})\mathrm{OPT}$ in polynomial time.

An almost tight example ($\sigma = 0, c >> \delta$)



Let T be a spanning tree with delay at most $(1 + \epsilon)$.

$$\lim_{k \to \infty} \frac{c(T)}{\mathrm{MST}} \nearrow 1 + \frac{1}{\epsilon}.$$

Proves tightness for shallow-light trees proposed in Cong et al. '92.

Thank you for your attention!