# Public Transport-based Crowd-shipping with Backup Transfers

Kerim U. Kızıl and Barış Yıldız

IPIC 2023

KOÇ UNIVERSITY

* This research has been supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under the grant number 218M605.
** The research of the second author has also been supported by Bilim Akademisi - The Science Academy, Turkey, under the BAGEP program.

# Introduction — Motivation of the Study

## Last Mile Delivery

- An integral part of any city logistics system
- Most expensive, time-consuming and operationally challenging part of the whole delivery process (Vanelslander et al., 2013)
- Conducted by traditional diesel trucks or vans
- With rising urbanization and exploding e-commerce, adds tremendous pressure on city logistics systems
- Classical last mile delivery systems cannot answer the needs of SMEs that wish to transform into e-tailers

Photo by STEPHEN VOSS
(Wall Street Journal: New Gridlock in America: The Fight for Curb Space)

## Introduction — Motivation of the Study (Cont'd)

Photo by Kldalley6 / CC BY-SA

Photo by Tony Webster / CC BY

**Several Ideas Proposed to Reform Last-Mile Delivery**

- Crowdshipping,
- Using public transport vehicles/infrastructure partly for freight,
- Deployment of parcel lockers (i.e., automated service points) in the strategic locations of urban areas,
- Replacing diesel trucks with other high-technology vehicles (e.g., drones, electric vehicles, delivery robots, etc.)

## Introduction — Proposed System

In this study, we propose a **new last-mile delivery model** in which several new delivery models and technologies are utilized together,

- (1) to make up for each other's shortcomings,
- (2) to boost each other's advantages.

**Collaborative effort of,**

- crowd,
- public transport (PT) vehicles,
- parcel lockers (i.e., service points),
- on-demand back-up delivery capacity offered by near-zero emission vehicles.

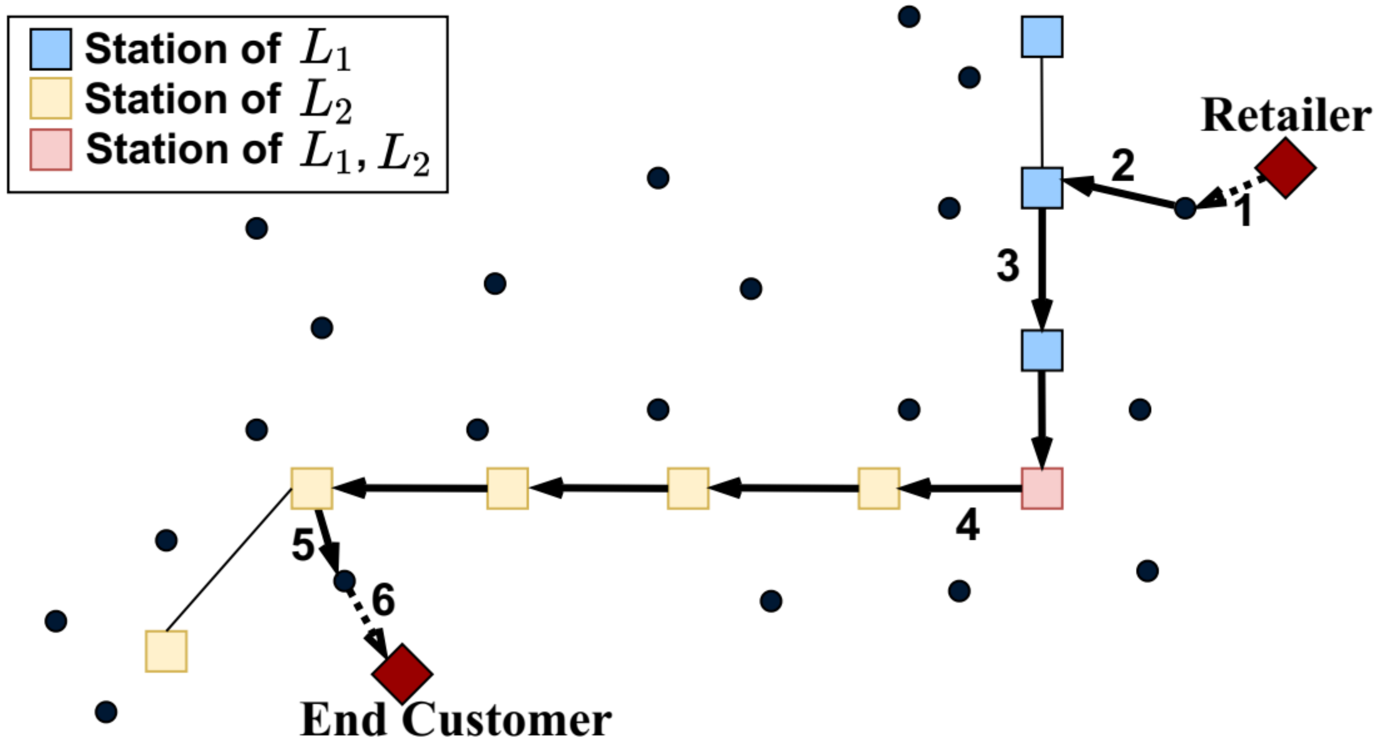
## Introduction — Proposed System (Cont'd)

**Details of the System**

- **Main objective:** To fulfill many-to-many express delivery demands (i.e., two or three-hour delivery requests) within an urban area.
- **Service points** are located in selected PT stations as well as several strategical points in the city. Service points in PT stations are referred to as *PT connections*. The remaining points are called *satellites*.
- **Use of Public Transit:** Transfers between PT connections are completed with PT vehicles.
- **Crowd-shippers:** travelers who are willing to insert package transfers into their already planned trips to earn a small compensation or to do good for the society/environment, take part in package transfers between the satellites and the PT connections.
- **Backup transfers:** Pick-up and deliveries by crowd-shippers and the use of backup transfers are organized in periodic distribution cycles (DC) to ensure timely deliveries.

## Introduction — Proposed System (Cont'd)

### A Practical Example



## Mathematical model

- $H$: set of candidate locations for PT connections
- $S$ : set of pick-up and delivery points, i.e., satellites
- $\Omega$: set of scenarios (realizations of demand and CS arrivals)

**Problem definition:**

PCBP is to find optimal locations for a given number of PT connections among a given set of alternatives $H$, such that the expected costs due to the use of backup transfers are minimized to carry out express shipments between a given set of satellites $S$, considering a given set of scenarios $\Omega$.

## Model Formulation

We formulate PCBP as an integer (binary) program with two sets of decision variables.

- The design variables $x_h,\ h \in H$ indicate whether to open a PT connection in location $h \in H$ or not (first-stage decisions),
- The routing variables $v_\pi^\omega,\ \pi \in \mathrm{P}, \omega \in \Omega$ indicate whether the backup transfer tour $\pi$ is used in scenario $\omega$ or not (second-stage decisions).

$$\min z = \sum_{\omega \in \Omega} \sum_{\pi \in P} p^\omega c_\pi v_\pi^\omega$$

# Model Formulation

$$(\mathcal{P})\quad \min z = \sum_{\omega\in\Omega}\sum_{\pi\in P} p^{\omega} c_{\pi} v_{\pi}^{\omega}$$

$$\text{s.t.}\quad \sum_{\pi\in P(s)} v_{\pi}^{\omega} + \sum_{h\in H(\omega,s)} x_h \geq 1, \qquad \forall\omega\in\Omega, \forall s\in S$$

Each satellite must be served in each scenario, either by CS or by backup transfers

$$\sum_{\pi\in\hat{P}(s,h)} v_{\pi}^{\omega} \leq x_h, \qquad \forall\omega\in\Omega, \forall h\in H, \forall s\in S$$

Do not start a backup transfer from a station that is not a PT connection

$$\sum_{\pi\in\hat{\hat{P}}(s,h)} v_{\pi}^{\omega} \leq x_h, \qquad \forall\omega\in\Omega, \forall h\in H, \forall s\in S$$

Do not end a backup transfer in a station that is not a PT connection

$$\sum_{h\in H} x_h = k$$

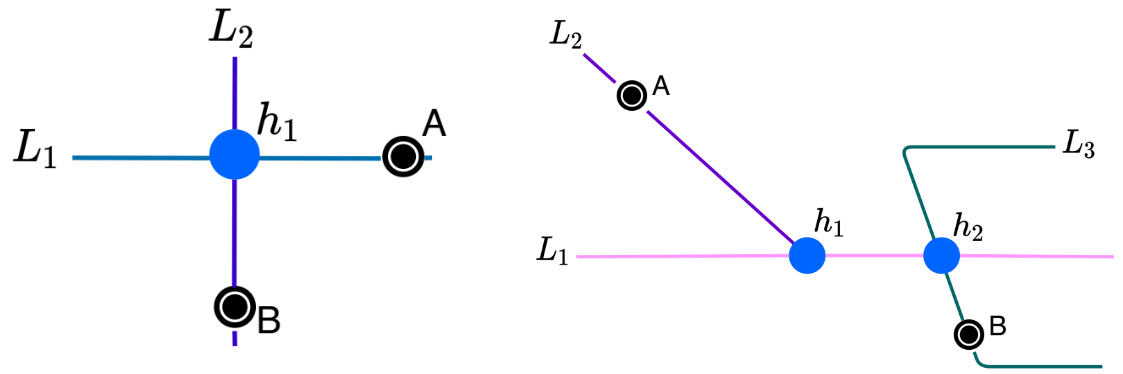
Exactly $k$ PT connections must be present in the system ($k$ being a pre-defined parameter)

$$x_h + x_m - x_n \leq 1, \qquad \forall h,m\in H, h\neq m, \forall n\in O(h,m)$$

Connectivity constraints

# Model Formulation

## Connectivity Constraints



The constraint

$$x_h + x_m - x_n \leq 1, \qquad \forall h, m \in H, h \neq m, \forall n \in O(h,m)$$

suffices if the PT network of focus is a tree (i.e., contains no cycles).

**Algorithm 2:** Simple Algorithm to Build the Sets $O(h,m)$

```
Input: $H$, PT lines
Output: $O(h,m),\ \forall h \in H, \forall m \in H,\ h \neq m$
1  $A$   // Creating an empty set to store $O(h,m)$ sets for different $h$ and $m$'s
2  $\hat{G} \leftarrow$ empty undirected graph
3  $\hat{G}$.addNodes(H)
4  foreach $h \in H$ do
5      foreach $m \in H$ do
6          if $h \neq m$ AND $h$ and $m$ are in the same PT line then
7              $\hat{G}.addEdge(\{h,m\})$
8  foreach $h \in H$ do
9      foreach $m \in H$ do
10         if $h \neq m$ then
11             $N \leftarrow$ depthFirst($h$, $m$)  // Suppose this calls depth first search
                algorithm that returns the path including origin and
                destination
12             $O(h,m) \leftarrow N \setminus \{h,m\}$
13             Add $O(h,m)$ to $A$
14 return $A$
```

## Branch-and-Price Algorithm (BP)

- Branch-and-price approach, at its essence, combines the branch-and-bound framework with a column generation procedure to add routing variables iteratively as needed (Barnhart et al., 1998).

- **Why use this algorithm?** The number of feasible routes becomes too large in realistic size problem instances, but only a few of them will be used in the optimal PCBP solution.

→ Solving LP (Column Generation Phase)

→ Solving IP (Branching Phase)

# Branch-and-Price Algorithm (BP) – Cont'd

## Solving LP (Column Generation Phase)

In $\mathcal{P}_{LP}^{k}$ for some $k \geq 0$, let $\delta$, $\phi$ and $\psi$ be the dual variables respectively associated with the constraints:

$$\sum_{\pi \in P(s)} v_\pi^\omega + \sum_{h \in H(\omega, s)} x_h \geq 1, \qquad \forall \omega \in \Omega, \forall s \in S$$

$$\sum_{\pi \in \hat{P}(s,h)} v_\pi^\omega \leq x_h, \qquad \forall \omega \in \Omega, \forall h \in H, \forall s \in S$$

$$\sum_{\pi \in \hat{\hat{P}}(s,h)} v_\pi^\omega \leq x_h, \qquad \forall \omega \in \Omega, \forall h \in H, \forall s \in S$$

Then, the reduced cost $\hat{c}_\pi^\omega$ of a routing variable $\mathbf{v}_\pi$ in scenario $\omega$ can be written as:

$$\hat{c}_\pi^\omega = p^\omega c_\pi - \sum_{s \in \ell_\pi} \left[\delta_s^\omega + \phi_{o_\pi s} + \psi_{d_\pi s}^\omega\right]$$

# Branch-and-Price Algorithm (BP) – Cont'd

**Solving LP (Column Generation Phase) – Cont'd**

To look for a path with negative reduced cost in a given scenario $\omega \in \Omega$, we use a label setting search algorithm (SA) that is equipped with several enhancement strategies to improve computational efficiency.

**Algorithm 1:** SA

```
Input: $\delta$, $\phi$, $\psi$, $H$, $\Omega$, $S$, m
Output: $V$

$V \leftarrow \{\}$
$\Gamma \leftarrow \{\gamma_h : h \in H\}$
do
    $\bar{\gamma} \leftarrow \arg\min_{\gamma \in \Gamma} \hat{c}_\gamma$
    Remove $\bar{\gamma}$ from $\Gamma$
    foreach $\omega \in \Omega$ do
        foreach $s \in S^{\omega}_{\bar{\gamma}}$ do
            Add $\bar{\gamma}_s$ to $\Gamma$
            foreach $h \in H_s$ do
                Let $\pi$ be a route with $o_\pi = h_{\bar{\gamma}_s}$, $\ell_\pi = \ell_{\bar{\gamma}_s}$ and $d_\pi = h$.
                if $\hat{c}^{\omega}_{\pi} < 0$ then
                    Add $v^{\omega}_{\pi}$ to $V$
while $\Gamma \neq \emptyset$ and $|V| < m$
return $V$
```

# Branch-and-Price Algorithm (BP) – Cont'd

## Solving IP (Branching Phase)

- In the study, we define branching rules for both design variables and routing variables.

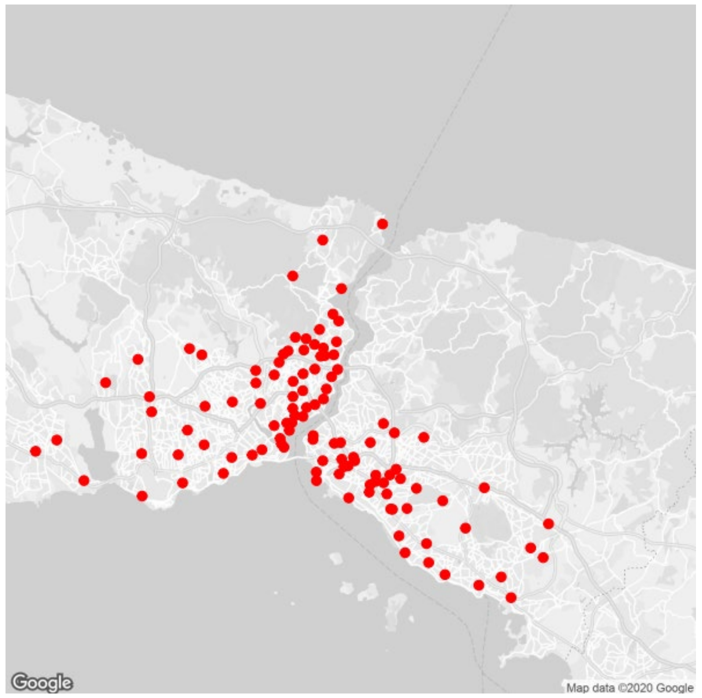
- Branching on design variables is usually enough for obtaining optimal solutions (as will be shown in the results section).

- Say, for a BP node, we finish the CG phase and find $x_{\bar{h}} \notin \{0, 1\}$ for $\bar{h} \in H$. Then, we branch on the design variable $x_{\bar{h}}$ with the following branching cuts:

$$x_{\bar{h}} \leq 0 \text{ and } x_{\bar{h}} \geq 1$$

# BP w/ Decomposition Branching (DB)

- As a recent innovation in mixed-integer programming, decomposition branching combines two essential techniques for solving mixed-integer programs (MIPs): **branch-and-bound** and **decomposition** (Yildiz et al., 2018).
- The main idea behind the suggested approach is to **consider the linking constraints as allocation of a "common resource" among the sub-problems** and build a branch-and-bound approach to find the **optimal allocation**.



# BP w/ Decomposition Branching (DB)

$$(\bar{\mathcal{P}}) \quad \min z = \sum_{i \in I} z_i$$

**s.t.**

$$z_i \geq \sum_{\omega \in \Omega} \sum_{\pi \in P(i)} p^{\omega} c_{\pi} v_{\pi}^{\omega}, \qquad \forall i \in I$$

$$\sum_{\pi \in P(i,s)} v_{\pi}^{\omega} + \sum_{h \in H(i,\omega,s)} x_h \geq 1, \qquad \forall \omega \in \Omega, \forall s \in S(i), \forall i \in I$$

Each satellite must be served in each scenario, either by CS or by backup transfers

$$\sum_{\pi \in \hat{P}(i,s,h)} v_{\pi}^{\omega} - x_h \leq 0, \qquad \forall \omega \in \Omega, \forall h \in H(i), \forall s \in S(i), \forall i \in I$$

Do not start a backup transfer from a station that is not a PT connection

$$\sum_{\pi \in \hat{\hat{P}}(i,s,h)} v_{\pi}^{\omega} - x_h \leq 0, \qquad \forall \omega \in \Omega, \forall h \in H(i), \forall s \in S(i), \forall i \in I$$

Do not end a backup transfer in a station that is not a PT connection

$$\sum_{i \in I} \sum_{h \in H(i)} x_h = k$$

Exactly $k_i$ PT connections must be present among the set $H(i)$

$$x_h + x_m - x_n \leq 1, \qquad \forall h, m \in H(i), h \neq m, \forall n \in O(h,m), \forall i \in I$$

Connectivity constraints

$x_h \in \{0,1\}, \forall h \in H(i), \forall i \in I$ and $v_{\pi}^{\omega}, \forall \omega \in \Omega, \pi \in P(i), \forall i \in I$ and $z_i \geq 0, \forall i \in I$

## BP w/ Decomposition Branching (DB)

In the new formulation, the reduced cost $\hat{\hat{c}}_{\pi}^{\omega}$ of a routing variable $\mathbf{v}_{\pi}$ in scenario $\omega$ can be written as:

$$\hat{\hat{c}}_{\pi}^{\omega} = \mu\, p^{\omega} c_{\pi} - \sum_{s \in \ell_{\pi}} \left[\delta_{s}^{\omega} + \phi_{o_{\pi}s} + \psi_{d_{\pi}s}^{\omega}\right]$$

The pricing problem defined for $\mathcal{P}$ continues to hold here with respect to its structure and solution methodology (i.e., exhaustive search algorithm). The only slight modification is on how we compute the reduced costs, which can be reflected into the algorithm in a quite straightforward way by replacing $\hat{c}_{\pi}^{\omega}$ by $\hat{\hat{c}}_{\pi}^{\omega}$ asgiven in equation above.

# BP w/ Decomposition Branching (DB)

$(DP_i(k_i)) \quad \min z_i$

**s.t.** $z_i \geq \sum_{\omega \in \Omega} \sum_{\pi \in P(i)} p^{\omega} c_{\pi} v_{\pi}^{\omega},$

$$\sum_{\pi \in P(i,s)} v_{\pi}^{\omega} + \sum_{h \in H(i,\omega,s)} x_h \geq 1, \qquad \forall \omega \in \Omega, \forall s \in S(i)$$

Each satellite must be served in each scenario, either by CS or by backup transfers

$$\sum_{\pi \in \hat{P}(i,s,h)} v_{\pi}^{\omega} - x_h \leq 0, \qquad \forall \omega \in \Omega, \forall h \in H(i), \forall s \in S(i)$$

Do not start a backup transfer from a station that is not a PT connection

$$\sum_{\pi \in \hat{\hat{P}}(i,s,h)} v_{\pi}^{\omega} - x_h \leq 0, \qquad \forall \omega \in \Omega, \forall h \in H(i), \forall s \in S(i)$$

Do not end a backup transfer in a station that is not a PT connection

$$\sum_{h \in H(i)} x_h = k_i$$

Exactly $k_i$ PT connections must be present among the set $H(i)$

$$x_h + x_m - x_n \leq 1, \qquad \forall h, m \in H(i), h \neq m, \forall n \in O(h,m)$$

Connectivity constraints

$x_h \in \{0,1\}, \forall h \in H(i)$ and $v_{\pi}^{\omega}, \forall \omega \in \Omega, \pi \in P(i)$ and $z_i \geq 0, \forall i \in I$

## BP w/ Decomposition Branching (DB)

### Branching Rules

At each branch-and-bound node with a solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{v}})$**,** we first check if the quantities $\hat{k}_i = \sum_{h \in H(i)} \bar{x}_h$ , $i = 1, 2, \dots, n$ are integers.

- If $\hat{k}_i$ is not integer for some $i = 1, 2, \dots, n$, we generate two branches with the following cuts.

$$\sum_{h \in H(i)} x_h \leq \lfloor \hat{k}_i \rfloor \wedge z_i \geq z^*_{i\lfloor \hat{k}_i \rfloor}$$

$$\sum_{h \in H(i)} x_h \geq \lceil \hat{k}_i \rceil$$

These cuts eliminate solutions with fractional number of PT connections in service region $\boldsymbol{i}$.

# BP w/ Decomposition Branching (DB)

## Branching Rules

- If $\hat{k}_i$ is all integers for all $i = 1, 2, \dots, n$, we start from the first service region (i.e., $i = 1$) and for each $i = 2, 3, \dots, n$ we check if the optimal solution of the subproblem $DP_i(\hat{k}_i)$ is larger than $\bar{z}_{ik_i} = \sum_{\omega \in \Omega} \sum_{\pi \in P(i)} p^{\omega} c_{\pi} \bar{v}_{\pi}^{\omega}$.

If we detect this is the case for some service region $i$, we generate two branches with the following cuts.

$$\sum_{h \in H(i)} x_h \le \hat{k}_i \wedge z_i \ge z^*_{i\hat{k}_i}$$

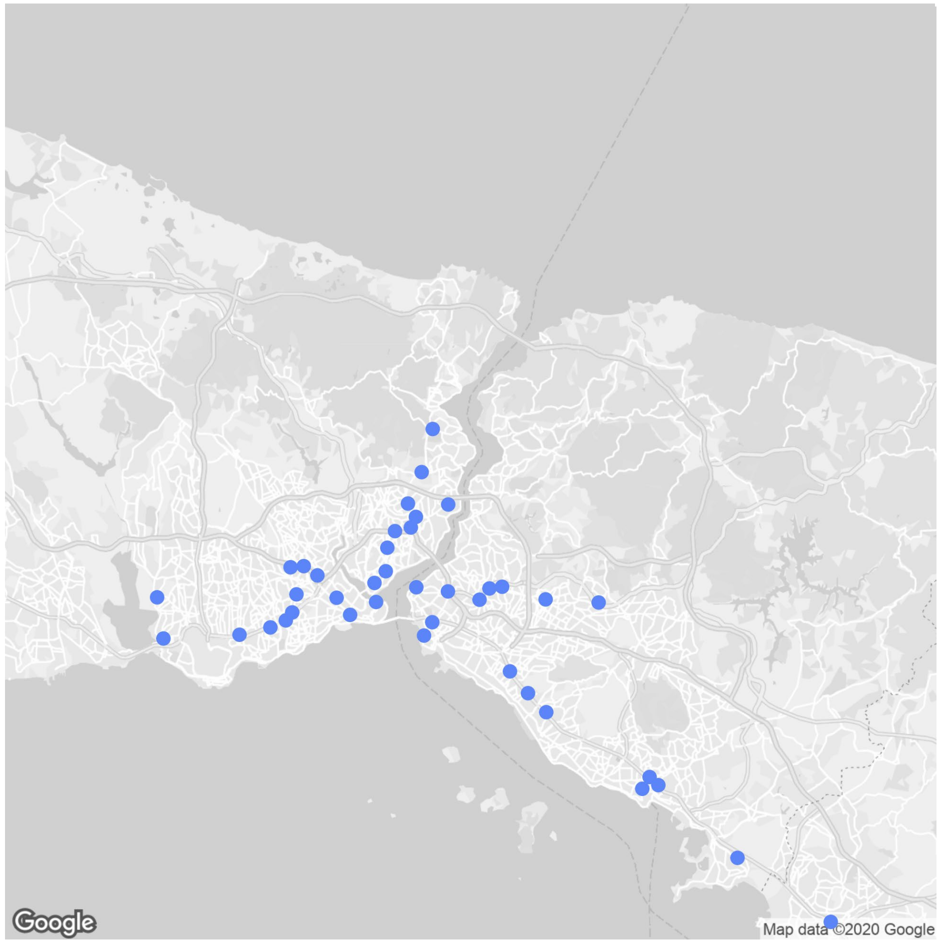
$$\sum_{h \in H(i)} x_h \ge \hat{k}_i + 1$$

These cuts eliminate solutions with fractional design variable values.

## Numerical Experiments

**Instance Generation**

- We study Istanbul and its railway-based public transportation system (RBPTS).
- We include seven RBPTS lines in our problem instances, namely, M1A, M2, M4, M5, M6, F1, and Marmaray (MA).
- Considered metro lines consist of 111 stations: 49 on the European and 62 on the Asian side. We select 39 of those stations as candidate locations for PT connections: 23 in the European and 17 on the Asian side.



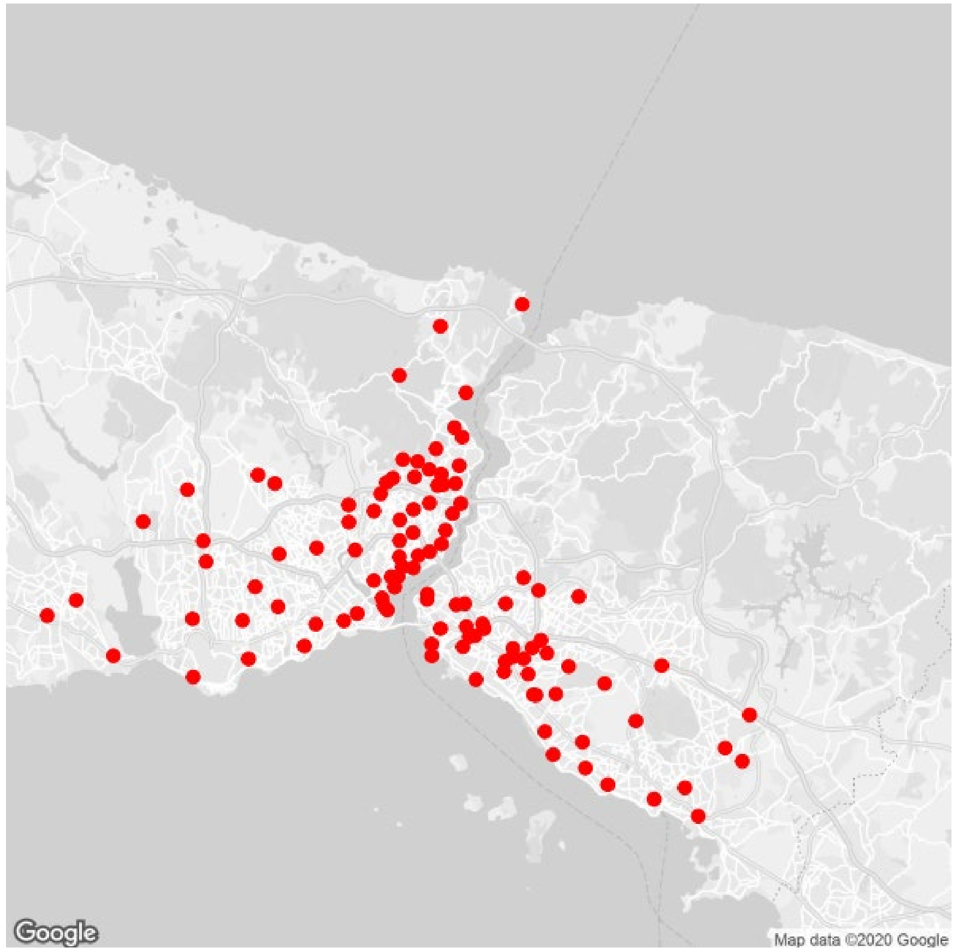
# Numerical Experiments

## Instance Generation – Cont'd

- As for the satellites, we choose 110 points (popular points of interest such as big supermarkets, shopping malls, universities, dormitories, schools, or some other central locations in neighborhoods). 65 of the 110 satellites reside on the European side, while the remaining 45 are on the Asian side.

- Ultimately, our instances consist of 150 vertices, 39 being candidate locations for PT connections and 110 being satellites.

- The distances between the vertices are obtained via *OpenRouteService* API and incorporated into our graph.



# Numerical Experiments

## Instance Generation – Cont'd

The set of scenarios we use in our instances are built as follows. To build a specific scenario $\omega$, we iterate over all pairs of candidate locations for PT connections and satellites, namely $\langle h, s\rangle, \forall h \in H, \forall s \in S$. For each pair $\langle h, s\rangle$ we draw a random number from a Bernoulli distribution with parameter $q$. If the drawn number is 1, we say opening a PT connection in $h$ enables CS for satellite $s$ in scenario $\omega$. The parameter $q$ is determined by the distance between $s$ and $h$ (in kilometers), denoted by $d(s,h)$ as well as a CS parameter $\zeta > 0$ as shown below.

$$q = \begin{cases} \min\{0.9\zeta, 1.0\}, & d(s,h) \leq 5 \\ \min\{0.8\zeta, 1.0\}, & 5 < d(s,h) \leq 10 \\ \min\{0.7\zeta, 1.0\}, & 10 < d(s,h) \leq 15 \\ \min\{0.6\zeta, 1.0\},, & 15 < d(s,h) \leq 20 \\ 0.0, & \text{otherwise} \end{cases}$$

# Numerical Experiments

## Experiment Design

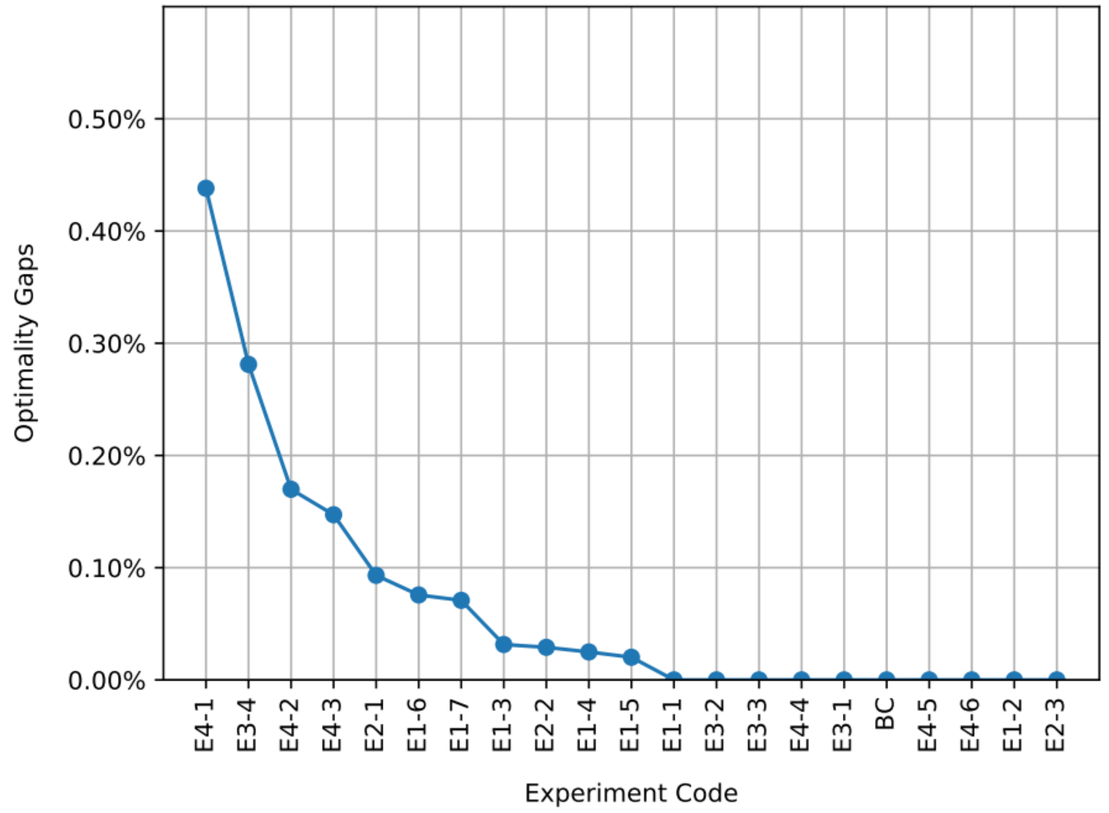
| Experiment Group | Parameter | Values |
|---|---|---|
| E1 | $\lvert\Omega\rvert$ | 10, 20, *30*, 40, 50, 60, 70, 80 |
| E2 | $t_{max}$ | *60*, 65, 80, 90 (min) |
| E3 | $k$ | 6, *8*, 10, 12, 14 |
| E4 | $\zeta$ | 0.7, 0.8, 0.9, *1.0*, 1.1, 1.2, 1.3 |

Base-case (BC) is characterized by $\lvert\Omega\rvert = 30$ , $t_{max} = 60\ min$, $k = 8$, $\zeta = 1.0$

Experiments in all groups are solved with plain BP as well as DB to investigate whether DB can provide any significant computational advantages.

# Results

**Optimality Gaps (branching solely on the design variables)**



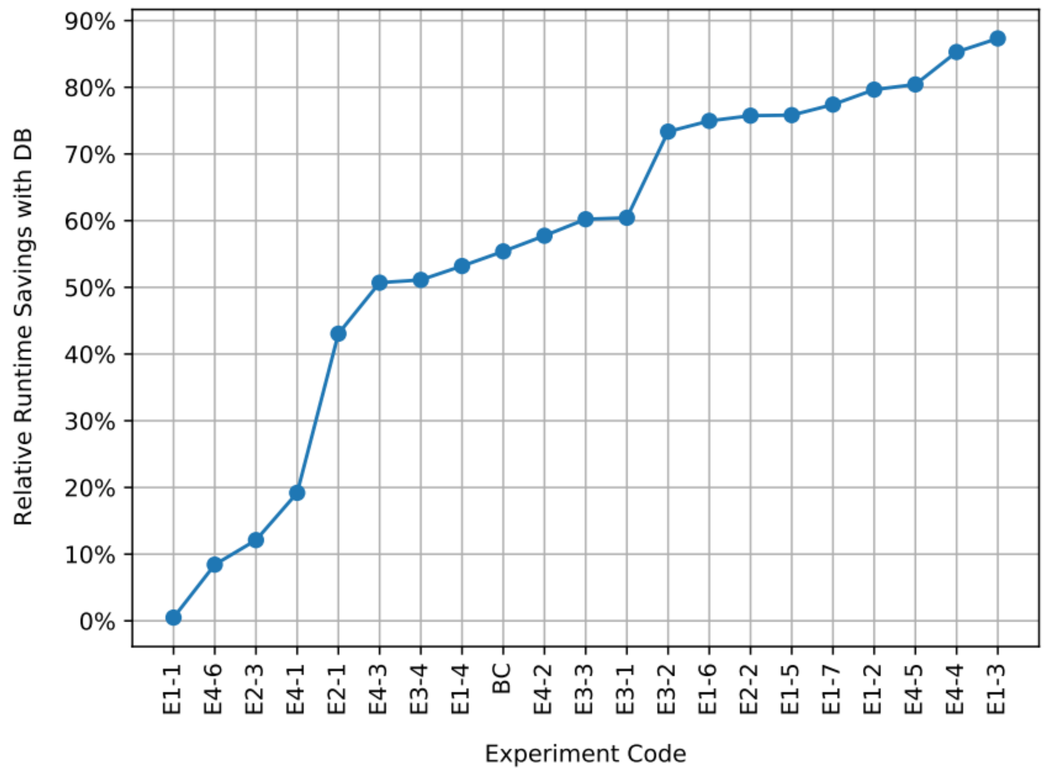
## Results

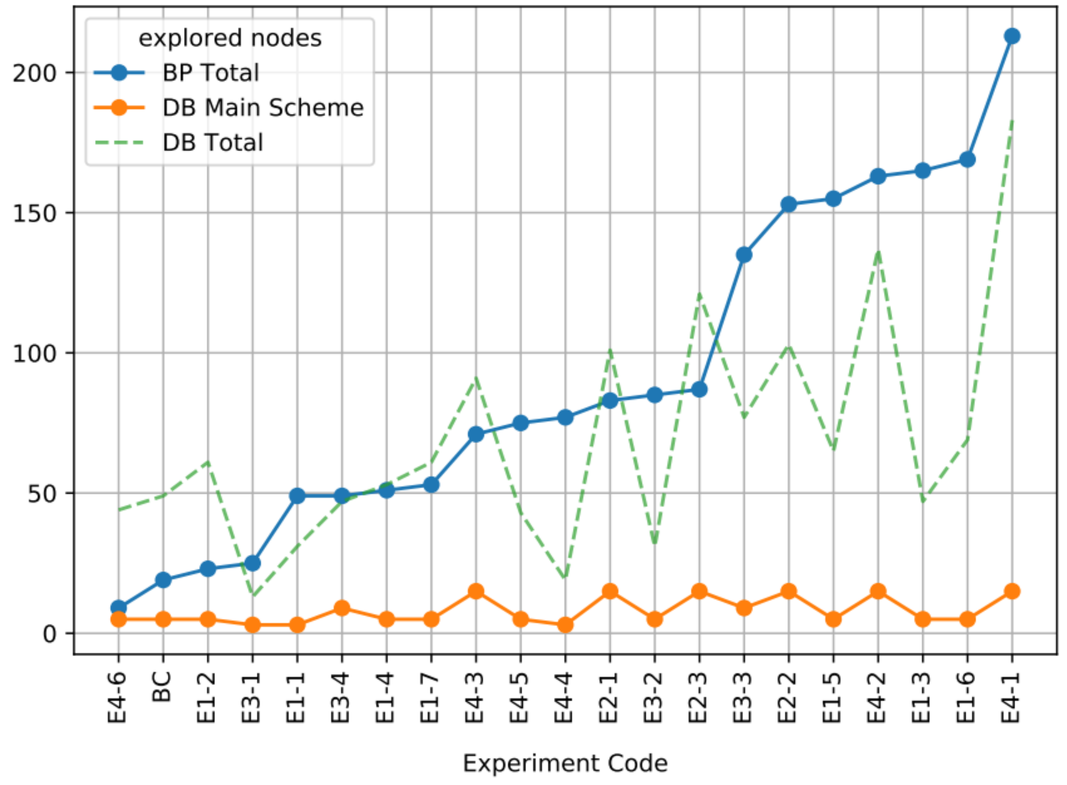
### Adequate Number of Scenarios

| $\|\Omega\|$ | $BP.time$ | $DB.time$ | $sol.val$ | $exp.cost$ | $PT.con$ |
|---|---|---|---|---|---|
| 10 | 654.70 | 177.48 | 330.36 | 426.44 | 0, 5, 14, 28, 38, 50, 85, 94 |
| 20 | 871.86 | 651.53 | 338.18 | 378.12 | 0, 5, 14, 28, 38, 50, 85, 94 |
| 30 | 2858.39 | 1275.00 | 378.36 | 378.12 | 0, 3, 14, 28, 38, 50, 86, 94 |
| 40 | 10427.67 | 1692.38 | 376.36 | 378.28 | 0, 3, 14, 28, 38, 50, 84, 94 |
| 50 | 13348.43 | 2605.49 | 381.62 | 378.28 | 0, 3, 14, 28, 38, 50, 84, 94 |
| 60 | 16359.74 | 3955.77 | 392.45 | 378.28 | 0, 3, 14, 28, 38, 50, 84, 94 |
| 70 | 21427.03 | 5365.68 | 401.95 | 378.12 | 0, 3, 14, 28, 38, 50, 86, 94 |
| 80 | 32556.96 | 7356.17 | 397.04 | 378.12 | 0, 3, 14, 28, 38, 50, 86, 94 |

# Results

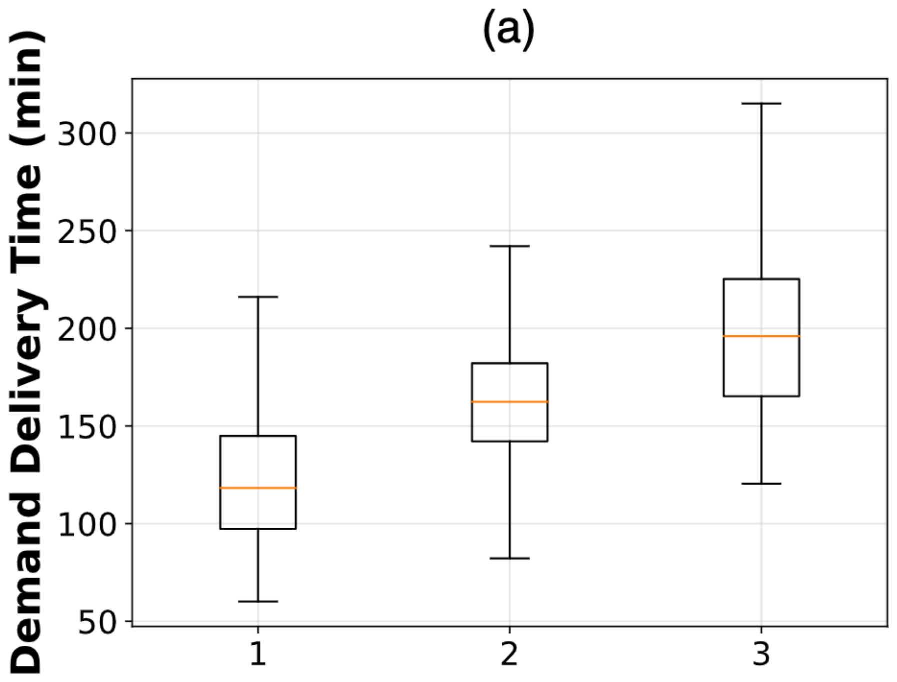
## BP vs DB: Computational Performance

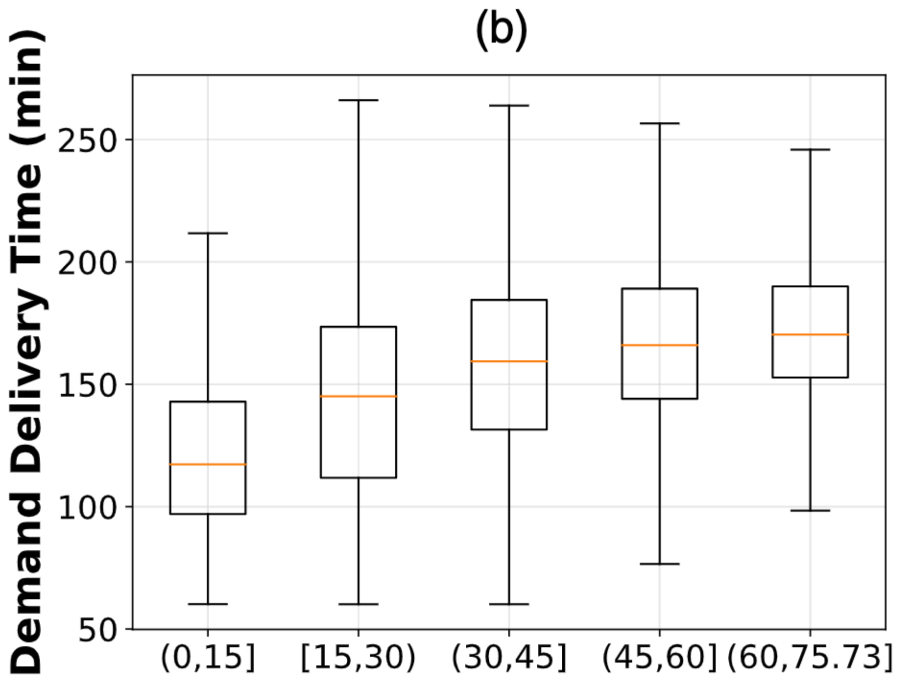




## Managerial Insights

**Delivery time distribution**

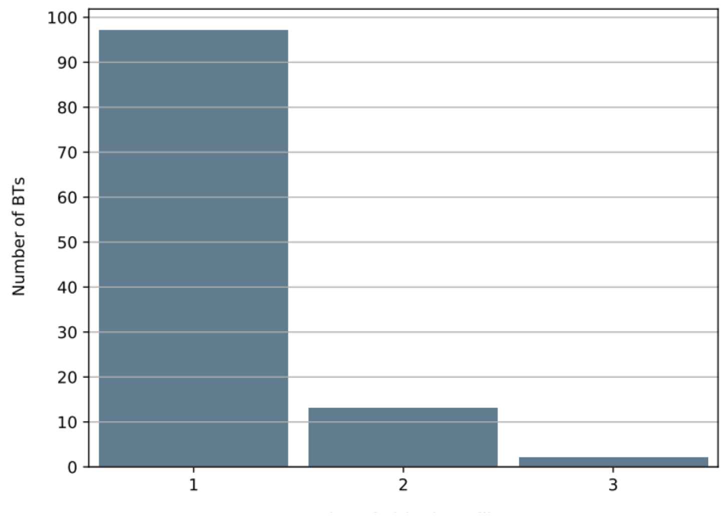


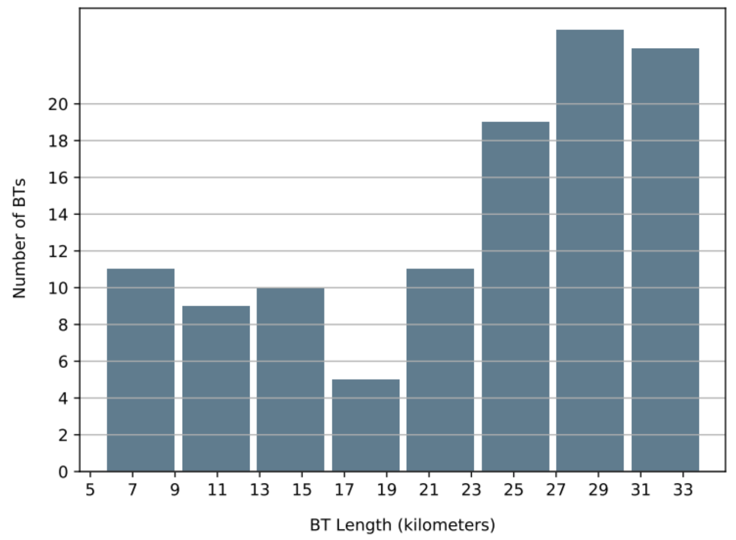


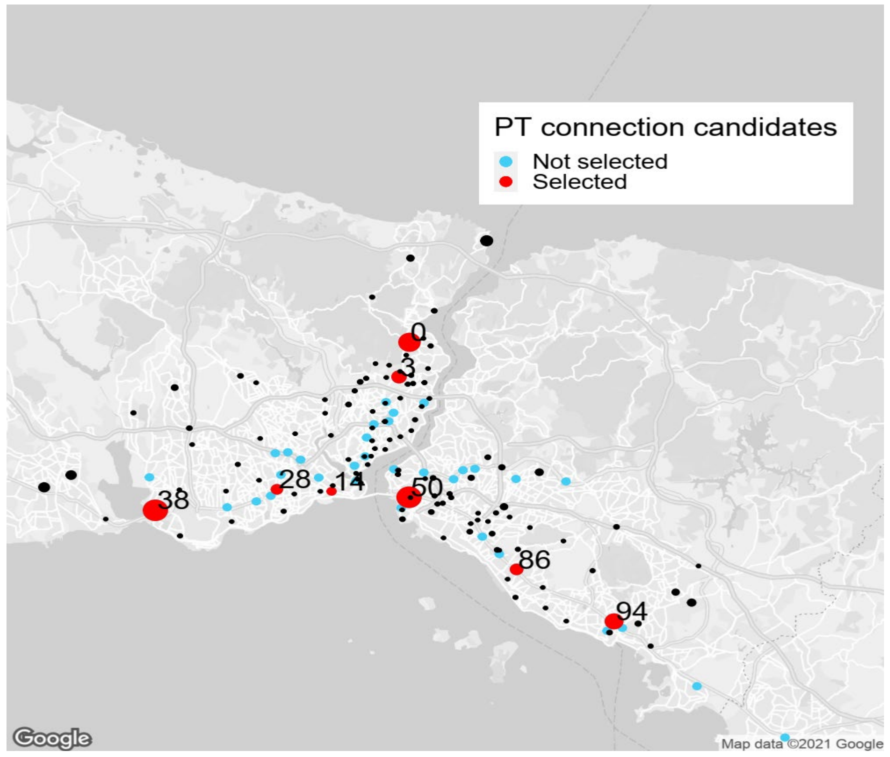
(a) Number of used lines. (b) Transfer distance (km).

# Managerial Insights

## Backup tour characteristics







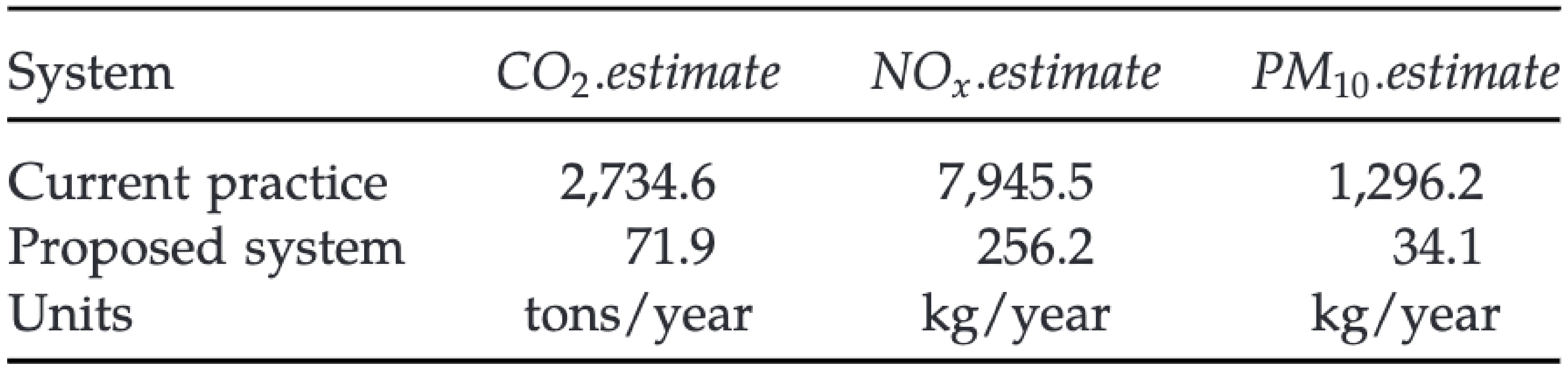
# Managerial Insights

**Current system**

- **445 diesel vans**
- **~27.000 km in traffic**
- **less than 1% same-day delivery**

**Proposed system**

- **6 electric vans**
- **less than 750 km in traffic**
- **more than 96.5% same-day delivery**



| System | $CO_2$.*estimate* | $NO_x$.*estimate* | $PM_{10}$.*estimate* |
|---|---|---|---|
| Current practice | 2,734.6 | 7,945.5 | 1,296.2 |
| Proposed system | 71.9 | 256.2 | 34.1 |
| Units | tons/year | kg/year | kg/year |

## Conclusion

- In this study, we present and investigate a novel public transport-based crowdshipping system
- We consider a scenario-based approach and develop a two-stage stochastic programming formulation to deal with the uncertainty in delivery capacity on top of the uncertainty associated with the demand.
- To solve this challenging problem for realistic size problem instances, we develop a branch-and-price algorithm.
- Taking advantage of a specific structure one can observe in many real-world applications, we explore the idea of boosting the computational performance of the basic BP algorithm by using a novel branching framework: decomposition branching.