Leveraging public transit for efficient last-mile delivery through crowdshipping

Mikele Gajda[a,], Filippo Ranza[b] Renata Mansini[b], Olivier Gallay[a]

[a]Department of Operations, HEC Lausanne, 1015 Lausanne, Switzerland
[b]University of Brescia, Italy

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

February 3, 2025

Keywords: Crowdshipping, Last-Mile Delivery, Public Transportation Network, Adaptive Large Neighborhood Search, Branch-and-Cut.

1 INTRODUCTION

Last-mile delivery, the final step in the supply chain, accounts for up to 28% of total delivery costs (Ranieri *et al.*, 2018). With the surge in e-commerce, it has become a significant challenge, contributing to urban congestion and pollution. Among innovative solutions, crowdshipping has emerged as a promising approach to address these issues (Archetti *et al.*, 2021, Arslan *et al.*, 2019, Filippi & Plebani, 2019). We introduce the Public Transport-based Crowdshipping Problem (PTCP), where public transport users act as crowdshippers (similarly to Wyrowski *et al.* (2024)), delivering parcels without detours or additional vehicles. In this system, a Logistics Service Provider (LSP) distributes parcels to lockers at specific stations within the public transport (PT) network, optimizing delivery based on crowdshipper availability, demand, costs, and synchronization requirements. Although these aspects are not individually new, this is the first model to integrate them all together in the same formulation. Our main contributions include:

- A novel mathematical formulation allowing multi-step deliveries and optimizing between crowdshipping and backup delivery.
- An enhanced formulation using valid inequalities and dynamic separation.
- An effective Adaptive Large Neighborhood Search (ALNS) metaheuristic to solve larger instances.
- Various problem instances reflecting realistic PT layouts.
- A sensitivity analysis revealing key success factors for PT-based crowdshipping.

This research advances the understanding of PT-based crowdshipping, offering insights into its potential to benefit urban logistics while addressing environmental concerns.

2 PROBLEM DESCRIPTION

Three key actors are involved in the PTCP: (i) the LSP, managing parcel delivery from the central warehouse; (ii) crowdshippers, represented by PT commuters available for parcel transfers between stations; and (iii) recipients, who select a subset of stations for parcel pickup. Figure 1 illustrates a PTCP example with three intersecting PT lines. To formalize this problem mathematically, we define a PT network as a set of PT lines $L = 1, ..., l_{max}$ and a set of stations N. In our example, $L = \{\text{red}, \text{green}, \text{blue}\}$, and $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. The subset $\bar{N} \in N$

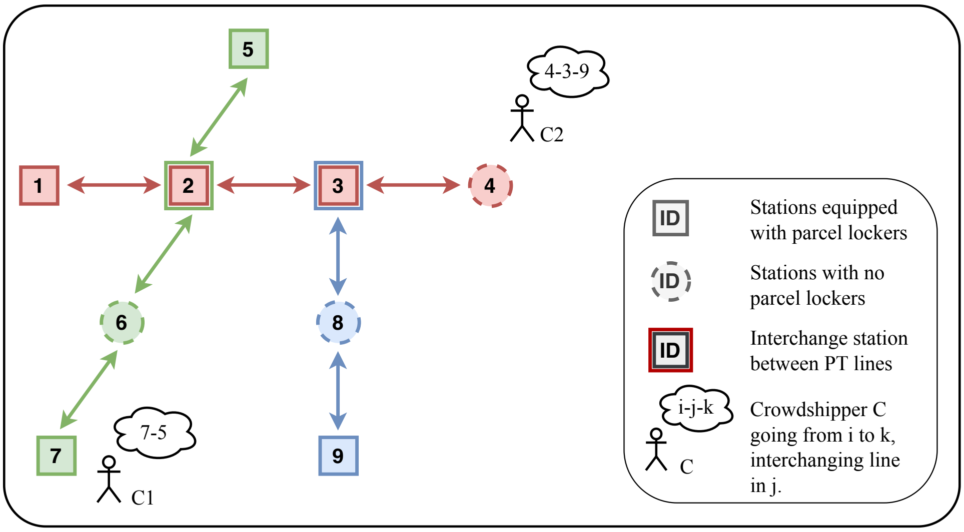

Figure 1 – *Example of the PTCP with three intersecting PT lines*

represents stations equipped with automated parcel lockers (with a given capacity), referred to as pick-up points (PuPs). In Figure 1, we can observe that not all stations have parcel lockers; for instance, stations 4, 6, and 8 lack this facility. For each line $l \in L$, $\bar{N}_l$ denotes the subset of PuP stations along that line. In our example: $\bar{N}_{\text{red}} = \{1, 2, 3\}$, $\bar{N}_{\text{green}} = \{5, 2, 7\}$, and $\bar{N}_{\text{blue}} = \{3, 9\}$. Here, stations 2 and 3 are respectively interchange stops between the red and the green line, and the red and the blue line. The network structure can be represented by a graph $\bar{G} = (\bar{N}, \bar{A})$, where $\bar{A} = \bigcup_{l \in L} \bar{A}_l$ and $\bar{A}_l = \{(i,j) | i, j \in \bar{N}_l\}$ represents all possible parcel transfers. Each arc $(i,j) \in \bar{A}$ has an associated travel time $\bar{t}_{ij}$. In our example, this would include arcs such as (1,2), (2,3), (5,2), (2,7), (3,9), etc. We indicate with K the set of crowdshippers, each following a predefined path $N_k \subset \bar{N}$. In Figure 1, we have two crowdshippers: C1 traveling from station 7 to station 5, and C2 traveling from station 4 to station 3 and then to station 9. We define $K_{ij} \subseteq K$ as the set of crowdshippers capable of transferring parcels between stations i and j. For instance, C1 $\in K_{7,5}$ and C2 $\in K_{4,3} \cap K_{3,9}$. The set of parcels is defined as P, with each parcel $p \in P$ associated with a set of possible destinations $D_p \subseteq D$ and a backup delivery cost γ_p. In our example, we have two parcels, p_i and p_j, with $D_{p_i} = 5$ and $D_{p_j} = 9$. To manage capacity constraints and synchronize transfers, we discretize time into slots $t \in T$. K_i^t denotes the set of crowdshippers at station i during time slot t. This is particularly important for coordinating transfers at interchange stations like 2 and 3 in our example. The objective of the PTCP is to optimize the assignment of parcels to crowdshippers, determine optimal routes, and select appropriate source and destination stations for parcels while minimizing total costs. These costs include crowdshipper remuneration ρ, source station activation costs ν_i, parcel loading costs σ_i, and backup delivery costs γ_p. In the proposed Mixed-Integer Linear Program (MILP), the objective function is stated as follows:

$$\min \; \rho \sum_{k \in K} z_k + \sum_{i \in S} \nu_i e_i + \sum_{i \in S} \sigma_i \sum_{p \in P} s_{ip} + \sum_{p \in P} \gamma_p q_p$$

Decision variables include z_k (crowdshipper selection), e_i (source station activation), s_{ip} (parcel-source assignment), and q_p (backup delivery use). In the context of our example from Figure 1, this objective function would help determine whether to use crowdshippers C1 and C2 (affecting z_k), which stations to use as sources (affecting e_i), how to assign parcels p_i and p_j to these sources (affecting s_{ip}), and whether to use backup delivery for either parcel (affecting q_p). The optimal solution would likely involve using crowdshippers to minimize costs, activating appropriate source stations (possibly stations 7 and 3), and avoiding the use of backup delivery if possible.

3 METHODOLOGY

Our approach to solving the PTCP involves three main components: a MILP formulation, a Branch-and-Cut framework with valid inequalities (both implemented and solved in CPLEX), and an ALNS algorithm within an Iterative Simulated Annealing (ISA) framework. We develop a strengthened MILP formulation for the PTCP, capturing all constraints and objectives of the problem. To strengthen this formulation, we introduce several families of valid inequalities within a Branch-and-Cut framework. These inequalities are dynamically added during the solution process and include constraints related to station activation, parcel-crowdshipper assignments, and location variables. The ALNS consists of destroy and repair operators. Destroy operators remove existing parcel routes and reallocate them to a backup delivery set, using four sorting rules for stations: Absolute Load, Relative Load, Average Load, and Random. Two main destroy operators are implemented: Parcel Paths Elimination and Capacity Reduction. Repair operators reconstruct solutions by selecting parcels from the backup delivery set and identifying relevant source stations. A tailor-made labeling algorithm, extending Dijkstra's algorithm, constructs feasible delivery paths for each parcel. Four repair operators are implemented, differing in their approach to parcel distribution and path construction strategies:
(`R1`) Parcel Distribution + Shortest Path
(`R2`) Delivery Cost Minimization + Shortest Path
(`R3`) Parcel Distribution + Uniform Crowdshipper Load
(`R4`) Delivery Cost Minimization + Uniform Crowdshipper Load
The ISA framework executes the ALNS multiple times, guided by a temperature schedule that controls the acceptance of worsening solutions. The initial solution is generated using a Constructive Initial Heuristic, employing the simplest repair heuristic (`R1`) with an "As-Is" parcel sorting strategy. By comparing the performance of the MILP formulation, the Branch-and-Cut algorithm, and the ALNS, we aim to provide insights into the most effective solution strategies for the PTCP under various problem sizes and characteristics

4 INSIGHTS ON RESULTS

Our study compared three approaches to solve the PTCP: Base CPLEX, CPLEX with valid inequalities in a Branch-and-Cut framework, and our ALNS algorithm. The following results provide quantitative insights into the effectiveness of these methods for various problem sizes. Base CPLEX demonstrated limited effectiveness, solving only 7 out of 144 instances (4.86%) to optimality within the given one-hour time limit. Even for the smallest instances ($|N| = 24$), average solution times exceeded 2,200 seconds. The solver often failed to explore beyond the root node, with an average MIP gap of 30.33%. The Branch-and-Cut algorithm with valid inequalities showed marginal improvements, being able to solve 8 instances to optimality (5.56%). While the time to best solution (TTB) increased by 4.65% on average, the mean MIP gap was reduced by 1.67%, from 30.33% to 28.66%. As predictable, given the complexity of the problem, the ALNS algorithm emerged as the most efficient approach, particularly for larger instances. Table 1 compares ALNS performance to CPLEX for the instances solved to optimality: For the nine instances solved to optimality by CPLEX, ALNS achieved optimal solutions in 6 out of 9 cases (66.67%) and near-optimal solutions (gap $< 1\%$) in the remaining 3, with an overall average gap of 0.097%. Crucially, ALNS accomplished this with an average computation time of 35.78 seconds, compared to CPLEX's 12,438 seconds. Additional computational tests show that for larger instances ALNS consistently outperforms CPLEX, with an average improvement of 9.48%. It is to be noted that the ALNS achieved this result within a 300-second time limit, while CPLEX utilized the full one-hour time limit. Parameter tuning analysis revealed optimal ALNS performance with cooling rates between 0.85 and 0.90, and runtimes exceeding 300 seconds. Statistical significance was confirmed using Wilcoxon signed-rank paired tests ($p < 0.05$).

Table 1 – *CPLEX and ALNS results on instances solved to optimality*

$\|N\|$	ID	CPLEX	ALNS	Gap (%)	CPLEX [s]	ALNS [s]
44	3	715	715	0.00	4944	1
40	6	614	614	0.00	21020	57
44	2	716	716	0.00	4233	1
24	3	638	638	0.00	2718	9
40	4	647	647	0.00	43932	233
40	1	750	750	0.00	96	0
44	5	684	685	0.15	2603	0
40	2	617	618	0.16	29475	20
40	5	708	712	0.56	5264	1

We additionally conducted a sensitivity analysis to evaluate how crowdshipping cost (cs-cost) and backup cost affect the delivery system's overall cost and efficiency. We varied cs-cost in $\{1, 2, 3\}$, backup cost ranges in $\{4-8, 6-12, 8-16\}$, number of crowdshippers in $\{15, 30, 45\}$, and the number of parcels in $\{(10, 20, 30\}$ across nine sets of instances.

The key findings and managerial insights are:

- Cost structure: Lower cs-cost and higher backup cost encourage crowdshipping. With cs-cost at 1 and backup cost in $8-16$, up to 81.81% of parcels use crowdshipping (40.31% on average). With cs-cost at 3 and backup cost in $4-8$, crowdshipping drops to 1.21% on average.
- Crowdshippers availability: More crowdshippers improve efficiency. For 10 parcels, increasing the number of available crowdshippers from 15 to 45 raises crowdshipping from 30% to 81.81% in the best case.
- Scalability: Efficiency decreases with volume. For 45 available crowdshippers, crowdshipping use drops from 81.81% (10-parcels scenario) to 43.51% (30-parcels scenario) in the best case.
- Time-slot configuration: Fewer time-slots impact performance. For the instances involving 30 and 45 parcels, costs increased by 9.11% when moving from 85 5-minute slots to 21 20-minute slots.

Ultimately, these quantitative insights provide useful directions for stakeholders to design and implement effective PT-based crowdshipping systems, balancing computational efficiency, cost optimization, and service quality.

References

Archetti, Claudia, Guerriero, Francesca, & Macrina, Giusy. 2021. The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, **127**, 105144.

Arslan, Alp M., Agatz, Niels, Kroon, Leo, & Zuidwijk, Rob. 2019. Crowdsourced Delivery—A Dynamic Pickup and Delivery Problem with Ad Hoc Drivers. *Transportation Science*, 222–235. Publisher: INFORMS.

Filippi, Carlo, & Plebani, Francesca. 2019. Metro stations as crowd-shipping catalysts: an empirical and computational study.

Ranieri, Luigi, Digiesi, Salvatore, Silvestri, Bartolomeo, & Roccotelli, Michele. 2018. A review of last mile logistics innovations in an externalities cost reduction vision. *Sustainability*, **10**(3), 782.

Wyrowski, Alexander, Boysen, Nils, Briskorn, Dirk, & Schwerdfeger, Stefan. 2024. Public transport crowdshipping: moving shipments among parcel lockers located at public transport stations. *OR Spectrum*.