

Column Generation for the Micro-Transit Zoning Problem

Hins Hu¹, Rishav Sen², Jose Paolo Talusan², Abhishek Dubey²,
Aron Laszka³ and Samitha Samaranayake¹

¹Cornell University

²Vanderbilt University

³Pennsylvania State University
zh223@cornell.edu

Abstract

Along with the rapid development of new urban mobility options like ride-sharing over the past decade, on-demand micro-transit services stand out as a middle ground, bridging the gap between fixed-line mass transit and single-request ride-hailing, balancing ridership maximization and travel time minimization. Micro-transit adoption can have significant social impact. It improves urban sustainability, through lower energy consumption and reduced emissions, while enhancing equitable mobility access for disadvantaged communities, thanks to its lower vehicle miles per passenger, flexible schedules, and affordable pricing. However, effective operation of micro-transit services requires planning geo-fenced zones in advance, which involves solving a challenging combinatorial optimization problem. Existing approaches enumerate candidate zones first and select a fixed number of optimal zones in the second step. In this paper, we generalize the Micro-Transit Zoning Problem (MZP) to allow a global budget rather than imposing a size limit on candidate zones. We also design a Column Generation (CG) framework to solve the problem and several pricing heuristics to accelerate computation. Extensive numerical experiments across major U.S. cities demonstrate that our approach produces higher-quality solutions more efficiently and scales better in the generalized setting.

1 Introduction

Public transit has long been fundamental to urban mobility systems, offering high energy efficiency [Hodges, 2010], reduced emissions, substantial capacity, and social equity benefits [Miller *et al.*, 2016]. However, its lengthy planning horizons, significant infrastructure investments, and budgetary constraints have resulted in limited coverage across most U.S. cities, leaving disadvantaged communities with poor accessibility to essential services and extended travel times. For example, according to the American Community Survey, only 3.7% of commuters in the U.S. use public transit for work commute in 2024.

On the other hand, on-demand ride-sharing has gained popularity as an alternative over the past decade, providing responsive door-to-door services [Kooti *et al.*, 2017]. Yet, it presents critical drawbacks: prohibitive costs for low-income commuters, higher per-passenger emissions, and insufficient pooling that increases deadheading, which exacerbates congestion and undermines sustainability.

Micro-transit has recently emerged as a middle ground between these two modes [Bardaka *et al.*, 2024]. This relatively high-capacity service mixes pre-booked and on-demand requests, and combines flexible routing and adaptive scheduling, filling coverage gaps left by fixed-route transit. It serves both short neighborhood trips and first-mile/last-mile connections to major hubs. With affordable pricing, micro-transit balances travel time, coverage, and cost across the urban mobility spectrum.

The first crucial technical challenge for city planners operating micro-transit is determining where to establish geo-fenced service areas given existing monetary budgets, specific requirements from local communities, and operational objectives (e.g., maximizing demand served). Micro-transit zoning design is a complex decision-making problem typically addressed by human experts without algorithmic processes or formal modeling. While various computational zoning approaches exist, many are adapted from different contexts, such as freight logistics [Chandra *et al.*, 2020] and urban development.

To the best of our knowledge, [Hu *et al.*, 2025] proposed the first optimal micro-transit zoning method to maximize total demand coverage. This two-phase framework develops an iterative algorithm with a geographical heuristic to enumerate all valid candidate zones under a size limit in Phase 1, then solves a Weighted Maximum Covering Problem via integer programming in Phase 2 to identify m optimal zones, where m is a problem input.

However, this work has notable limitations: (1) the zoning model assumes prior knowledge of the number of zones to establish (i.e., m is a fixed input), rather than allowing a global budget for operating all zones within an urban region’s boundary, which is less realistic; (2) the enumeration algorithm in Phase 1 suffers from scalability issues in larger cities when demand aggregation becomes more granular (e.g., less than 1 km²).

In this paper, we propose a more efficient approach using

Column Generation (CG) to address a generalized zoning design setting with a *global budget*, which we call the Micro-Transit Zoning Problem (MZIP). Rather than relying solely on geographical heuristics for candidate zone generation, we reformulate the problem into a CG framework consisting of a *restricted master problem* and a *pricing problem*, both modeled as Integer Linear Programs (ILPs). Under the CG framework, we leverage mathematical programming to handle exponentially many decision variables, aligning with the nature that the candidate set of valid zones is exponentially large. To further accelerate computation, we develop a pricing heuristic to replace the exact pricing ILP with negligible compromise in solution quality. The effectiveness of our framework is validated through extensive experiments across multiple large U.S. cities. Numerical results demonstrate the superiority of the proposed CG-based approach over the method by [Hu *et al.*, 2025]. Additional analysis and parameter tuning for our method are provided to better guide engineering practice.

While our main experimental results focus on demand maximization, the proposed framework also admits an equity-aware objective extension described in Section 4.6.

1.1 Stakeholder Collaboration and Societal Impact

This research addresses a real-world societal challenge: the need for computationally tractable and equitable approaches to design micro-transit service zones for cities that face low transit accessibility and financial burdens to expand their fixed-line transit networks. The work emerges from an ongoing multi-year collaboration with the Chattanooga Area Regional Transportation Authority (CARTA), our primary stakeholder and domain partner, which serves a region where only 1.6% of commuters currently use public transit for work commutes, significantly below the national average.

CARTA actively participates in this project through three primary roles: (1) conducting community surveys and providing insights for algorithm design; (2) coordinating with other local stakeholders, such as social workers and mini-bus drivers, to calibrate our model parameters; (3) leading a micro-transit pilot project for real-world validation.

1.2 Contributions

Our contributions to the research community of AI for social good are threefold:

1. **Generalized Problem and Formulation:** We generalize the Micro-Transit Zoning Problem (MZIP) to incorporate a global budget constraint on operational costs and reformulate it as a Column Generation (CG) framework.
2. **Scalable Pricing Heuristic:** We develop an efficient pricing heuristic algorithm for scalability.
3. **Real-World Validation:** We conduct extensive experiments across multiple large U.S. cities using real-world mobility demand data, demonstrating superior performance over state-of-the-art methods in both solution quality and computational efficiency.

2 Related Work

In this section, we first review the existing computational methods for zoning in urban planning and transportation, including ones particularly for micro-transit services. Second, we briefly discuss the development of Column Generation (CG) and its applications on relevant problems in transportation, such as ride-sharing and vehicle routing.

2.1 Zoning in Urban Planning and Transportation

The design of spatial zones plays a critical role in urban planning and infrastructure development, shaping land use decisions [Lens, 2022] and influencing the trajectory of urban growth [Domingo *et al.*, 2021]. Inadequately designed or obsolete zoning can intensify spatial inequalities in cities [Dingil, 2025]. In the domain of general transportation planning, a great number of computational methods have emerged to support automated or optimal zoning. For example, approaches have been developed using K-means clustering [Chandra *et al.*, 2020] or multi-objective genetic algorithms [Chandra *et al.*, 2021] to refine zone structures by grouping geographically similar spatial units.

In the specific context of micro-transit zoning, computational methods remain limited. For example, [Liu and Ouyang, 2021] proposed a framework integrating fixed-route transit with zone-based demand-responsive services, where the zone size is optimized jointly with transit network design. However, their approach assumes uniform zones adjacent to each other in a continuous region. [Bonner and Miller-Hooks, 2023] developed a mixed-integer programming approach for equitable micro-transit zone generation, prioritizing fair service distribution across populations and geographic areas. Most relevant to our work, [Hu *et al.*, 2025] formulated the micro-transit zoning problem to maximize demand coverage and proposed a two-phase solution framework involving candidate zone generation and optimal selection. As stated in Section 1, our work extends their problem setting and provides a more scalable solution approach based on Column Generation.

2.2 Column Generation

Column Generation (CG) emerged in the 1950s as a technique for solving large-scale Linear Programs (LPs), where explicitly enumerating all decision variables is computationally prohibitive. The seminal work of [Dantzig and Wolfe, 1960] established its theoretical foundations. Early successes in the Cutting Stock Problem [Gilmore and Gomory, 1961] demonstrated its practical value. The integration of CG with branch-and-bound led to the development of branch-and-price algorithms [Barnhart *et al.*, 1998], enabling the solution of large-scale Integer Programs (IPs). As of now, CG has become a cornerstone methodology in artificial intelligence, particularly for problems with decomposable structures. Further details can be found in [Desaulniers *et al.*, 2006].

CG has been widely applied across various fields, including vehicle routing, scheduling, resource allocation, ride-sharing, and more. [Desrochers *et al.*, 1992] solved the Vehicle Routing Problem with Time Windows using a set-partitioning formulation where each column represents a

feasible route, and the pricing problem is formulated as a resource-constrained shortest path problem. [Vance *et al.*, 1997] applied CG to the Airline Crew Pairing Problem, generating crew pairings (i.e., columns) that cover flight segments while satisfying labor regulations - a method now widely adopted in the airline industry. [Van Den Akker *et al.*, 1999] employed CG for the Parallel Machine Scheduling Problem, where columns represent feasible job sequences for individual machines. [Riley *et al.*, 2019] developed a real-time share-a-ride system using column generation to handle up to 30,000 ride requests per hour in New York City. Our paper contributes to this repertoire by applying CG to the Micro-Transit Zoning Problem.

3 Problem Statement

We define the **Micro-Transit Zoning Problem (MZP)** as a combinatorial optimization problem with the following inputs:

1. A set of uniform, non-overlapping cells V in an urban region, with each cell representing a small neighborhood that aggregates all travel demand from and to the area. Non-adjacent cells are allowed if no travel demand arises in the area in between.
2. A road network G in the same region where V lies on.
3. A demand table $d : V^2 \rightarrow \mathbb{R}_+$, where each entry $d(i, j)$ represents a nominal travel demand from cell i to j .
4. A micro-transit zone $S \in 2^V$ is defined as a subset of cells that are *adjacent* to each other in the *road network space*, not in the 2-D geographical space.
5. A distance matrix $c : V^2 \rightarrow \mathbb{R}_+$, where each entry $c(i, j)$ denotes the shortest-path distance in the network G from the centroid of cell i to the centroid of cell j .
6. A linear cost function $f(S) = \alpha D_S^2 + \beta$ that maps the *squared diameter* D_S^2 of a zone S to the cost of operating that particular zone. The diameter is defined as the distance between the most distant two cells within a zone, namely $D_S \triangleq \max_{i, j \in S} c(i, j)$.
7. A global budget $B \in \mathbb{R}_+$ for total cost of operating all micro-transit zones.

The objective is to select a collection of micro-transit zones to operate under budget B , so as to maximize the total intra-zone demand shown in Equation (1).

$$\sum_{i, j \in V} d(i, j) \mathbb{I}(i, j \text{ in the same zone}) \quad (1)$$

The problem studied here generalizes that of [Hu *et al.*, 2025]. Specifically, we model the monetary constraint of transit agencies as a global budget for all operated zones collectively, rather than imposing a uniform size limit on each individual zone, resulting in a more realistic setting.

For input 1, non-uniform cells can be designed to match demand patterns without affecting our approach. Input 6 reflects our partner transit agency's insight that operational cost is proportional to area size. Beyond the global budget (input 7), a single-zone budget can also be incorporated if needed.

4 Methodology

In this section, we briefly introduce Column Generation (CG) and its key terminology, then describe each component as applied to our micro-transit zoning problem.

4.1 Prerequisite for Column Generation

CG is a powerful decomposition technique particularly effective for problems that can be formulated as *set partitioning* or *set covering* models, where each decision variable represents a partial solution with special structure - referred to as a *column*.

Due to the exponentially growing nature of feasible columns with the problem size, CG works on a *Restricted Master Problem (RMP)*, which has the same structure as the original problem but includes only a subset of columns. Starting with an initial small subset, CG iteratively identifies and adds promising columns to the RMP.

As shown in Figure 1, the process alternates between two steps. First, the RMP is solved to obtain an optimal solution and associated *dual variables*. Second, these duals are passed as input to the *Pricing Problem* - an auxiliary optimization problem that searches for columns with negative *reduced cost*.¹ Any column with negative reduced cost indicates a potential increase in the RMP objective and is added to the column set.

CG terminates when the pricing problem no longer identifies improving columns, guaranteeing optimality of the linear relaxation to the unrestricted master problem. To obtain integer solutions, column generation is typically embedded within a branch-and-bound framework, known as *branch-and-price* [Barnhart *et al.*, 1998], where column generation solves the LP relaxation at each node of the search tree.

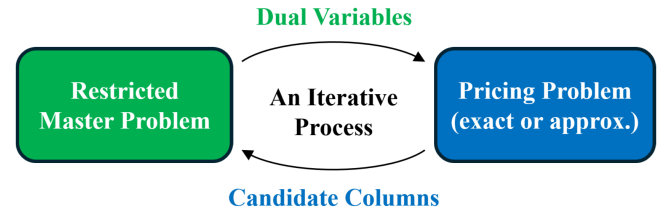


Figure 1: A Sketch of Column Generation Process

4.2 The Restricted Master Problem

As discussed in Section 4.1, applying CG requires a set-based formulation. In the micro-transit zoning problem, each column naturally corresponds to a zone as it is a collection of cells, which is analogous to vehicle routes serving as columns in the Vehicle Routing Problems [Desrochers *et al.*, 1992].

In addition to the notations introduced in Section 3, we further define the following for the formulation of the CG framework. Let S be the set of candidate zones. Let $x_S \in \{0, 1\}$ be a binary decision variable indicating whether to select zone S . Let $w_{ij} \in \{0, 1\}$ be an auxiliary decision variable indicating whether two distinct cells i and j are in the same zone, which is used to avoid double-counting the demand whose

¹ If the RMP is a maximization, the pricing aims for columns with positive reduced cost, which is exactly our case.

origin and destination are both in the overlapping area of two distinct zones. Let $z_S = [z_S^1, \dots, z_S^i, \dots, z_S^{|V|}]^\top$ be the cell-selection vector of zone S , where z_S^i is a binary coefficient indicating whether cell i is included in zone S . Then, we formulate the RMP as in ILP (2).

$$\max_{\mathbf{x}, \mathbf{w}} \sum_{i,j \in V} d_{ij} w_{ij} \quad (2a)$$

$$\text{s.t.} \quad \sum_{S \in \mathcal{S}} (\alpha D_S^2 + \beta) x_S \leq B \quad (\lambda) \quad (2b)$$

$$w_{ij} - \sum_{S \in \mathcal{S}} z_S^i z_S^j x_S \leq 0 \quad (\pi_{ij}) \quad \forall i, j \in V \quad (2c)$$

$$x_S \in \{0, 1\} \quad \forall S \in \mathcal{S} \quad (2d)$$

$$w_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (2e)$$

Constraint (2b) requires that the cumulative operational cost of all selected zones be capped by the global budget. Constraint (2c) links $\mathbf{x}$ and $\mathbf{w}$, forcing w_{ij} to be 0 if cells i and j are not assigned simultaneously to any zone S (with at least one of z_S^i, z_S^j , and x_S being 0). In the maximization setting, it also raises w_{ij} to 1 if cells i and j are assigned to at least one zone (with $z_S^i = 1, z_S^j = 1$, and $x_S = 1$ for at least one $S \in \mathcal{S}$). Constraints (2d) and (2e) enforce the integrality of decision variables. For better readability, the dual variables λ and π_{ij} are marked next to their corresponding constraints.

Despite its theoretical soundness, the above formulation often exhibits degeneracy in real-world instances, which can cause the CG process to stall or cycle [Du Merle *et al.*, 1999]. As an engineering practice, we perturb Constraint (2c) by adding small random noise terms ϵ_{ij} to the right-hand side, which breaks degeneracy at the cost of non-monotonic convergence in the master objective value.

$$w_{ij} - \sum_{S \in \mathcal{S}} z_S^i z_S^j x_S \leq \epsilon_{ij} \quad \forall i, j \in V \quad (3)$$

According to [Bertsimas and Tsitsiklis, 1997], the reduced cost for zone S can be computed by Equation (4), which will be used in the pricing problem for determining whether S can be added to the candidate column set.

$$\bar{c}_S = \sum_{i,j \in V} \pi_{ij} z_S^i z_S^j - \lambda (\alpha D_S^2 + \beta) \quad (4)$$

ILP (2) shares the structure of ZONINGILP in [Hu *et al.*, 2025] except for adopting a global budget. Notably, the candidate zone set $\mathcal{S}$ in our CG framework is orders of magnitude smaller, as dual variables guide the pricing problem to generate only “premium” candidate zones.

4.3 The Exact Pricing Problem

In the pricing problem, the core idea is to find a zone to maximize the reduced cost, and thus the cell selection vector z_S becomes part of the decision and the index S for zones can be discarded. Formally, we define a binary decision variable z_i for every cell $i \in V$ to indicate whether i is selected in the zone.

Similarly, we define a positive continuous decision variable $D^2 \in \mathbb{R}_+$, with the index S discarded as well, to represent the actual “squared diameter” of the selected zone. This quantity is an input to the RMP because the set of candidate zones is given. However, in the pricing problem, it varies in the optimization process guided by the dual variables.

Then, an exact Integer Quadratic Programming (IQP) formulation for pricing is shown in IQP (5). From Constraint (5b), as long as both cell i and j are selected simultaneously in a zone, its diameter is lower-bounded by the two-way shortest path distance between them. The max operator is used to account for the asymmetry in city road networks. The objective function is adopted from the reduced cost in Equation (4) by converting $\mathbf{z}$ and D to decision variables and dropping the constant term $\lambda \beta$.

$$\max_{\mathbf{z}, D^2} \sum_{i,j \in V} \pi_{ij} z_i z_j - \lambda \alpha D^2 \quad (5a)$$

$$\text{s.t.} \quad D^2 \geq \max\{c^2(i, j), c^2(j, i)\} z_i z_j \quad \forall i, j \in V \quad (5b)$$

$$z_i \in \{0, 1\} \quad \forall i \in V \quad (5c)$$

Since a generic IQP can be computationally challenging for modern mixed-integer programming solvers, we further linearize the bilinear term ($z_i z_j$) to obtain an ILP pricing problem as an alternative. Let y_{ij} be a binary decision variable for each pair of cells (i, j) , indicating whether both cells are selected. This definition coincides with that of w in the master problem (Section 4.2), although the variables serve different roles. Then, the linearized formulation is shown in ILP (6).

$$\max_{\mathbf{y}, \mathbf{z}, D^2} \sum_{i,j \in V} \pi_{ij} y_{ij} - \lambda \alpha D^2 \quad (6a)$$

$$\text{s.t.} \quad y_{ij} \leq z_i \quad \forall i, j \in V \quad (6b)$$

$$y_{ij} \leq z_j \quad \forall i, j \in V \quad (6c)$$

$$y_{ij} \geq z_i + z_j - 1 \quad \forall i, j \in V \quad (6d)$$

$$D^2 \geq \max\{c^2(i, j), c^2(j, i)\} y_{ij} \quad \forall i, j \in V \quad (6e)$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (6f)$$

$$z_i \in \{0, 1\} \quad \forall i \in V \quad (6g)$$

4.4 Pricing Heuristic

While the exact pricing formulations in IQP (5) and ILP (6) guarantee optimality, they can still be computationally intractable for large-scale instances, particularly when the pricing problem must be solved repeatedly throughout the CG process. To ensure the effectiveness of the framework, we develop a *construction heuristic* that rapidly generates high-quality zones with positive reduced cost.

Motivated by the generation of school bus routes in [Bertsimas *et al.*, 2019], our heuristic employs a greedy node-addition strategy: starting from a randomly selected pair of cells (which should satisfy the single-zone budget constraint if enforced), it iteratively adds cells that maximize the marginal gain in reduced cost until no beneficial addition remain. The marginal gain of adding cell k to the current zone

S can be efficiently computed by Equation (7).

$$\Delta(k) = \sum_{j \in S} (\pi_{kj} + \pi_{jk}) - \lambda \cdot \alpha \cdot (D_{\text{new}}^2 - D_{\text{current}}^2) \quad (7)$$

where D_{new} represents the updated squared diameter after including cell k , and D_{current} is the current diameter of zone S . The first term captures the added demand coverage from new within-zone connections, while the second term accounts for the increased operational cost due to zone expansion.

To enhance robustness and exploration, the heuristic can be executed multiple times with different random initialization. Each run produces a local optimal zone from a particular seed pair of nodes. Upon termination, all generated zones with positive reduced costs are added to the RMP's candidate set S . Algorithm 1 presents the complete procedure.

Algorithm 1: Greedy Pricing with Random Seeds

Input: cell set V , distances c , duals λ and $\{\pi_{ij}\}$, parameters α and β , number of runs R , single-zone budget B_0

Output: list of zones $\mathcal{Z}$ with positive reduced cost $\mathcal{Z} \leftarrow \emptyset$,

for $r = 1$ **to** R **do**

// Random Initialization

repeat

Randomly select two distinct cells $i, j \in V$

$D \leftarrow \max\{c[i, j], c[j, i]\}$

until $\alpha \cdot D^2 + \beta \leq B_0$;

$S \leftarrow \{i, j\}$

// Greedy Node Addition

repeat

$k^* \leftarrow \arg \max_{k \in V \setminus S} \Delta(k)$

$\hat{D} \leftarrow \max_{j \in S} \max\{c[k^*, j], c[j, k^*]\}$

$D_{\text{new}} \leftarrow \max\{D, \hat{D}\}$

if $\Delta(k^*) > 0$ **and** $\alpha \cdot D_{\text{new}}^2 + \beta \leq B_0$ **then**

$S \leftarrow S \cup \{k^*\}$

$D \leftarrow D_{\text{new}}$

else

break

until *convergence or timeout*;

// Zone Selection

Compute $\bar{c}_S = \sum_{i, j \in S} \pi_{ij} - \lambda(\alpha D^2 + \beta)$

if $\bar{c}_S > 0$ **then**

$\mathcal{Z} \leftarrow \mathcal{Z} \cup \{S\}$

return $\mathcal{Z}$

The heuristic offers several fundamental advantages. First, it executes in $O(|V|^2)$ time per random run, significantly faster than solving the exact pricing problem, either in the formulation of ILP or IQP. Second, it increases column diversity through random initialization, thus lowering the risk of the CG process being trapped in local optima. Third, the heuristic supports a time limit, functioning as an *anytime* algorithm that enables flexible trade-offs between efficiency and solution quality.

4.5 The Overall Framework

We summarize the complete CG framework for the micro-transit zoning problem, as presented in Algorithm 2.

Algorithm 2: CG for Micro-Transit Zoning

Input: cell set V , demand d , distances c , parameters α and β , budget B

Output: Selected zones S^*

// Initial Candidate Set

$S \leftarrow$ a random zone under single-zone budget B_0

// Column Generation

repeat

$(\lambda, \{\pi_{ij}\}) \leftarrow \text{RMP}(S)$

if *exact* **then**

$S_{\text{new}} \leftarrow \text{EXACTPRICING}(\lambda, \{\pi_{ij}\}, \dots)$

else

$S_{\text{new}} \leftarrow \text{PRICINGHEURISTIC}(\lambda, \{\pi_{ij}\}, \dots)$

$S \leftarrow S \cup S_{\text{new}}$

until $S_{\text{new}} = \emptyset$ *or timeout*;

// Solve Restricted Master

$S^* \leftarrow \text{RMP}(S)$

return S^*

Our framework only applies CG in the LP relaxation to the RMP (i.e., the root node in the branch-n-bound tree), without embedding the procedure into a more sophisticated full branch-n-price scheme. While the latter would be theoretically required to guarantee optimality, its implementation would demand substantial re-engineering of lower-level, fine-tuned branching rules and cutting plane algorithms encapsulated in any modern Mixed Integer Programming (MIP) solver, making it an impractical option.

Nevertheless, through the numerical experiments in Section 5, we show that our framework is sufficiently superior to the state-of-the-art baseline.

After the solver returns the selected zones, we further apply a cost-preserving post-processing step to absorb omitted cells without increasing any zone cost. The pseudocode (Algorithm 3) and rationale are provided in A.

4.6 Equity-Aware Objective Extension

We also consider an equity-aware extension to better prioritize underserved communities. For each H3 cell i , we define a cell-level equity score e_i . Specifically, we combine a no-vehicle score $\widetilde{VE}_i = 1 - VE_i$ and a low-income score $\widetilde{IC}_i = 1 - \text{rank}(IC_i)$ into

$$e_i = \mu \widetilde{VE}_i + (1 - \mu) \widetilde{IC}_i, \quad (8)$$

where $\mu \in [0, 1]$ is a weighting parameter and $\text{rank}(IC_i) \in [0, 1]$ is the city-wide percentile rank of median household income of cell i . The underlying vehicle ownership and income attributes are mapped from ACS 5-year census block-group data [U.S. Census Bureau, 2026] to H3 cells by area-weighted interpolation. For each OD pair (i, j) , we use the origin-cell score e_i and replace the RMP objective with

$$\max_{x, w} \sum_{i, j \in V} e_i d_{ij} w_{ij}. \quad (9)$$

This modifies only the objective coefficients. The constraint structure, CG decomposition, and exact or heuristic pricing procedures remain unchanged.

5 Numerical Experiments

In this section, we present comprehensive experiments to evaluate the effectiveness and scalability of our CG framework on real-world micro-transit zoning instances. We consider five medium-to-large U.S. cities: Miami, Boston, Atlanta, Chattanooga, and Nashville. The code and implementation are publicly available on GitHub.

5.1 Data Preparation

City boundaries and underlying road networks are publicly available. For Chattanooga, travel demand is derived from a proprietary dataset of real origin-destination (OD) trips provided by our partner transit agency, CARTA. For all other cities, demand is synthetically generated following the methodology in [Sen *et al.*, 2025], which leverages two real-world sources: (1) departure time and travel time distributions from the American Community Survey (ACS), and (2) residence-to-workplace flows from the Longitudinal Employer-Household Dynamics (LODES). In pre-processing, we apply a geo-fencing filter to retain only trips with both origins and destinations inside the city boundaries and exclude very short trips with total travel distance less than 500 meters.

City	# of H3 Hexagons		Road Network	
	Res. 7	Res. 8	# of nodes	# of edges
Miami	31	220	9086	23558
Boston	45	321	11408	26086
Atlanta	73	502	12905	33628
Chattanooga	80	551	9486	24358
Nashville	257	1803	22470	55796

Table 1: H3 Hexagon and Road Network Statistics Across Multiple Cities

To spatially aggregate travel demand into small, contiguous neighborhoods - each corresponding to a cell as defined in Section 3 - we use the open-source Hexagonal Hierarchical Spatial Index (H3) developed by Uber. H3 divides geographic space into hexagonal grids at multiple discrete resolutions. We focus on resolution 7 (average cell length ≈ 1.22 km) and resolution 8 (≈ 0.46 km), which provide a reasonable granularity for micro-transit zoning based on insights from CARTA. Trips are assigned to hexagons by mapping their origin and destination coordinates to the corresponding cells. As an example, the aggregated demand pattern in Chattanooga is visualized in Figure 2.

In summary, geographic statistics for all cities are reported in Table 1. The total demand for all city instances (res-7 and res-8) is reported in B.

5.2 Experimental Settings

The experiments are conducted under the following configurations and parameter settings unless otherwise specified. We

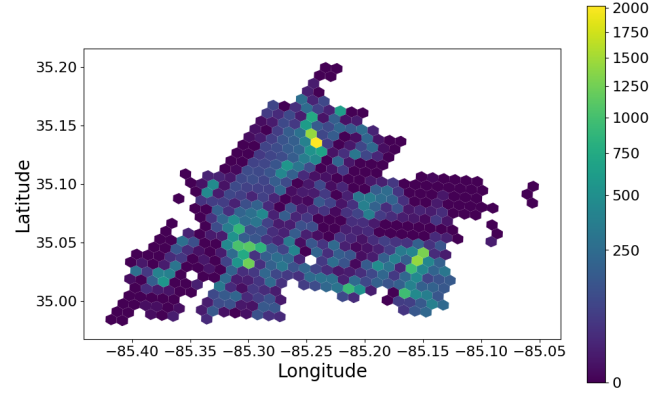


Figure 2: Aggregated number of trips in Chattanooga with H3 Resolution 8. The value for each cell counts both trips from and trips ending in that cell.

use pre-computed nominal driving times on the real-world road networks to construct the distance matrix c . To make operational costs comparable across cities of different sizes, distances are normalized to the range $[0, 1]$.

The cost parameters are set uniformly across all cities: global budget $B = 8$, cost scale factor $\alpha = 5$, and fixed cost per zone $\beta = 1$. To prevent zones from becoming excessively large, reflecting the design of micro-transit systems as a complement rather than a replacement for public transit, we set a single-zone budget of $B_0 = 2$. Under this configuration, optimal solutions are expected to contain three to five zones, each less than the size of one-fifth of a city. While the cost structure and coefficients are artificial, we are collaborating with our partner transit agency, CARTA, to calibrate these parameters in an ongoing micro-transit pilot deployment.

Experiments are run individually on a c6a.xlarge AWS instance with 4 CPUs and 8 GiB memory. For exact pricing, the ILP (6) is solved with a 10-hour limit for CG and 1 hour per pricing problem. For the pricing heuristic, CG is limited to 20 minutes in total, and 2 minutes per pricing problem. The number of random runs R in Algorithm 1 is set to 10. All mathematical programs involved are solved by Gurobi.

5.3 Main Results

The baseline algorithm is from [Hu *et al.*, 2025], hereafter referred to as CLIQUEGEN. Because the original implementation does not incorporate a global budget B and requires specifying the number of zones m , we modify it to compute the operational cost of any candidate zone, defined as $\alpha \cdot D^2 + \beta$, alongside the clique construction process.

We first compare the computational times between CLIQUEGEN and CG. From Table 2, we can see that CLIQUEGEN only completes successfully for Miami and Boston with resolution-7 H3 hexagons. For all larger instances, CLIQUEGEN terminates with an out-of-memory error in less than 1 hour on the dedicated AWS instance. In contrast, CG + EXACTPRICING successfully completed all runs within the 10-hour time limit. When the problem size scales from Miami to Boston, the computational time for the CG framework increases by only 3x (from 0.16 to 0.49 sec-

onds), while CLIQUEGEN increases by more than 100x (from 0.95 to 106 seconds). This dramatic difference demonstrates that CLIQUEGEN suffers from severe scalability issues and is not effective for large cities.

Instances	CLIQUEGEN	CG + EXACT PRICING
Miami-7	0.95	0.16
Boston-7	106	0.49
Atlanta-7	OOM	7.56
Chattanooga-7	OOM	9.94
Others	OOM	Completed

Table 2: Computational times in seconds between CLIQUEGEN and CG + EXACTPRICING across all {city, H3 resolution} configurations. OOM means the out-of-memory error.

Since both CLIQUEGEN and CG + PRICINGHEURISTIC are anytime algorithms, we impose the same time limit of 1200 seconds on both methods and compare the best solutions obtained across all large instances with more than 200 hexagons. To ensure robustness, the results from CG are averaged over 5 independent experiments. Figure 3 demonstrates that CG + PRICINGHEURISTIC significantly outperforms CLIQUEGEN, achieving approximately 36.7% higher demand coverage on average.

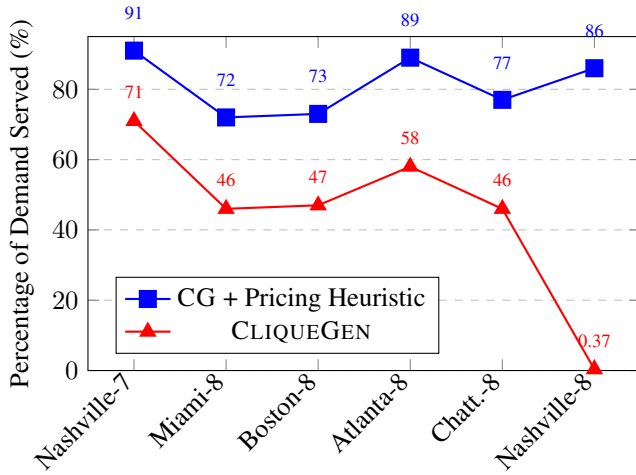


Figure 3: Comparison of demand coverage between CLIQUEGEN and CG + PRICINGHEURISTIC within a 20-minute compute budget

The performance gap is particularly pronounced in the largest instance: Nashville at resolution-8 with 1803 hexagons. Here, CG + PRICINGHEURISTIC leverages dual information to guide its search, exploring approximately 50 to 60 high-quality candidate zones and achieving 86% demand coverage. In stark contrast, CLIQUEGEN generates over 227,106 very small zones due to its iterative construction of cliques from smaller cliques, resulting in a highly unstable outcome with only 0.37% demand coverage.

This dramatic distinction highlights a fundamental advantage of CG’s dual-driven strategy: it uses economic signals

from dual prices to prioritize zones that maximize marginal contribution to demand coverage.

As an illustration, Figure 4 visualizes the zones selected by CG + PRICINGHEURISTIC for Chattanooga at resolution 8, which largely cover the high-demand areas shown in Figure 2.

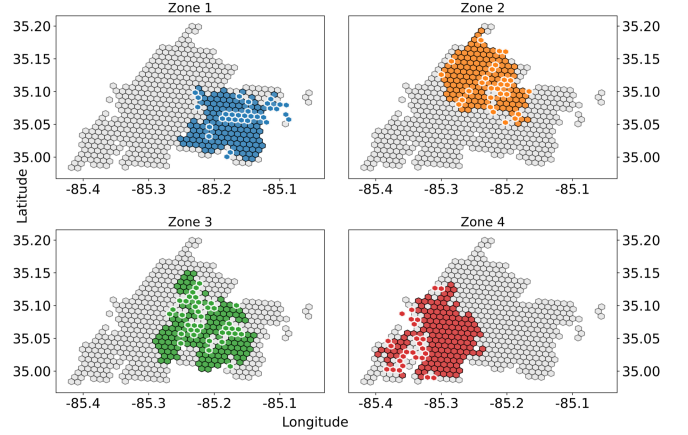


Figure 4: The micro-transit zones selected by CG + PRICINGHEURISTIC in Chattanooga at H3 resolution 8. Cells with white bold outlines are patched by the cost-preserving post-processing algorithm in A.

5.4 Exact vs. Heuristic Pricing

We also compare the exact ILP formulation (ILP 6) and the heuristic (Algorithm 1) for solving the pricing problem. As shown in Table 3, both approaches achieve nearly identical solution quality across the selected large instances. Except for Nashville at resolution 8, where the exact ILP times out without a result, they perform similarly in all other instances. Given that the pricing heuristic runs approximately 30x faster than the exact method (from their time limits in Section 5.2), it emerges as the preferred option for practical implementation.

Instances	Exact ILP	Pricing Heuristic
Boston-8	74.32%	72.84%
Chattanooga-8	76.80%	76.59%
Miami-8	79.92%	72.49%
Atlanta-8	88.58%	89.06%
Nashville-7	91.34%	91.12%
Nashville-8	Timeout	85.74%

Table 3: Comparison of demand coverage between Exact ILP formulation and pricing heuristic across five cities

5.5 ILP vs. IQP in Exact Pricing

We also compare the average computational times between the ILP formulation (6) and the IQP formulation (5) for solving the pricing problem. As shown in Table 4, the IQP formulation consistently outperforms the ILP by 2x to 4x across all instances, despite quadratic programs being generally harder

to solve. This gap likely stems from the excessive number of pairwise variables y_{ij} introduced to the ILP, making the model less compact.

	Miami-8	Boston-8	Atlanta-8	Chatt.-8	Nash.-8
ILP	350	486	1236	>3600	>3600
IQP	95	185	532	2016	>3600

Table 4: Average computation time in seconds for exact ILP and IQP pricing on selected large instances. All observations are capped by the 3600 seconds time limit.

5.6 Equity-Aware Zoning in Chattanooga

As a brief case study, we compare the original demand-maximizing objective and the equity-aware objective in Chattanooga under the same configuration. For a percentile threshold i , we define *high equity-score demand coverage* $HESDC(i)$ as the percentage of served demand originating from cells with equity scores (defined in Section 4.6) at or above the i -th percentile. At $i = 80$, the equity-aware objective increases $HESDC(80)$ from 81.01% to 82.29%. At the same time, the total raw demand served decreases by 1.65% on average. This suggests that the framework can improve coverage for high equity-score cells with only a modest efficiency loss.

Figure 5 summarizes the improvement in $HESDC(i)$ for $i \in \{70, 75, 80, 85, 90\}$. The improvement remains positive across all tested percentiles of equity scores, ranging from 1.28% to 2.34%. The results also suggest an approximately increasing improvement as the percentile threshold becomes higher. The total demand served is unchanged across these threshold evaluations because they are computed on the same fixed zoning solutions rather than by re-optimizing the model for each threshold. The results in this section are averaged over 10 independent random runs.

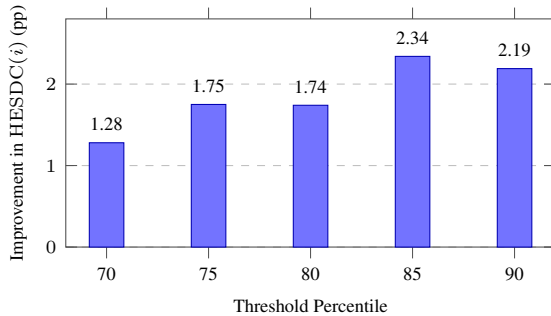


Figure 5: Ten-run average improvement in $HESDC(i)$ from the equity-aware objective in Chattanooga.

6 Conclusion and Future Direction

This paper proposes a Column Generation-based framework for the Micro-Transit Zoning Problem and demonstrates its

effectiveness through extensive experiments on multiple large U.S. cities. Our solution approach, informed by practical insights from local city planners and transit agencies, such as our partner CARTA, has the potential to generate significant social impact by providing more sustainable and equitable micro-transit services.

Future work will consider a more general zoning model that jointly accounts for intra-zone and inter-zone demand, with cross-zone travel served by fixed-route transit, enabling micro-transit to operate as an effective first-and-last-mile connector to mass public transportation.

A Cost-Preserving Zone Patching

After the CG solver returns the selected zones, we apply a lightweight post-processing step over the full cell set V to patch additional cells into each zone. This covers both zero-demand cells that regularize the zone shape and support potential future demand, and demand-carrying cells that may have been omitted due to heuristic suboptimality. These cells are identified as *admissible cells* (Definition 1).

Definition 1. For a zone S , let $D_S = \max_{i,j \in S} c(i, j)$ denote its diameter, as defined in Section 3. A cell $g \in V \setminus S$ is an admissible cell for zone S if $\max_{j \in S} \max\{c(g, j), c(j, g)\} \leq D_S$.

Since only admissible cells are added, the zone diameter and therefore the operating cost $f(S) = \alpha D_S^2 + \beta$ remain unchanged. The pseudocode for this patching procedure is given in Algorithm 3.

Algorithm 3: Cost-Preserving Zone Patching

Input: Selected zones S^* , cell set V , distances c

Output: Patched zones $\hat{S}^*$

```

 $\hat{S}^* \leftarrow S^*$ ;
foreach  $S \in \hat{S}^*$  do
     $D_S \leftarrow \max_{i,j \in S} \max\{c(i, j), c(j, i)\}$ ;
    repeat
        foreach  $g \in V \setminus S$  do
            if  $\max_{j \in S} \max\{c(g, j), c(j, g)\} \leq D_S$  then
                 $S \leftarrow S \cup \{g\}$ ; // cost unchanged
    until  $S$  unchanged;
return  $\hat{S}^*$ 

```

B Total Demand Across Cities

The total demand is reported in Table 5.

City	Resolution 7	Resolution 8
Chattanooga	30480	31779
Miami	67828	73013
Atlanta	120316	124421
Boston	207140	273272
Nashville	290014	291139

Table 5: Total demand (number of O-D trips) for H3 resolutions 7 and 8 across the cities.

Ethical Statement

This work studies computational tools for micro-transit zoning to support more sustainable and equitable urban mobility. The experiments use aggregate mobility and geographic data and do not involve personally identifiable information. Since zoning decisions may affect service allocation across communities, the proposed methods should be used as decision-support tools together with stakeholder engagement, local policy judgment, and fairness checks, rather than as fully automated decision-making systems.

Acknowledgements

This work was supported in part by a Cloud Computing Grant from Amazon Web Services (AWS), by Cornell Advanced Computing, and by the U.S. Department of Energy (DOE) under Award No. DE-EE0011188. We also thank Hongyi Jiang for helpful brainstorming and conceptual discussions on the methodology and algorithms.

Contribution Statement

Hins Hu contributed to all aspects of the project, including formulation, methodology, implementation, experiments, analysis, and writing. Rishav Sen and Jose Paolo Talusan contributed to data preparation and the data pipeline. Abhishek Dubey, Aron Laszka, and Samitha Samaranayake contributed to supervision and funding acquisition. Samitha Samaranayake provided primary technical supervision, and Aron Laszka contributed to manuscript writing and revision.

References

- [Bardaka *et al.*, 2024] Eleni Bardaka, Pascal Van Hentenryck, Crystal Chen Lee, Christopher B Mayhorn, Kai Monast, Samitha Samaranayake, and Munindar P Singh. Empathy and ai: Achieving equitable microtransit for underserved communities. *International Joint Conferences on Artificial Intelligence Organization*, 2024.
- [Barnhart *et al.*, 1998] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.
- [Bertsimas and Tsitsiklis, 1997] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA, 1997.
- [Bertsimas *et al.*, 2019] Dimitris Bertsimas, Arthur Delarue, and Sebastien Martin. Optimizing schools’ start time and bus routes. *Proceedings of the National Academy of Sciences*, 116(13):5943–5948, 2019.
- [Bonner and Miller-Hooks, 2023] Taylor Bonner and Elise Miller-Hooks. Achieving equitable outcomes through optimal design in the development of microtransit zones. *Journal of Transport Geography*, 112:103696, 2023.
- [Chandra *et al.*, 2020] Aithichya Chandra, Agnivesh Pani, and Prasanta K Sahu. Designing zoning systems for freight transportation planning: a gis-based approach for automated zone design using public data sources. *Transportation Research Procedia*, 48:605–619, 2020.
- [Chandra *et al.*, 2021] Aitichya Chandra, MN Sharath, Agnivesh Pani, and Prasanta K Sahu. A multi-objective genetic algorithm approach to design optimal zoning systems for freight transportation planning. *Journal of Transport Geography*, 92:103037, 2021.
- [Dantzig and Wolfe, 1960] George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960.
- [Desaulniers *et al.*, 2006] Guy Desaulniers, Jacques Desrosiers, and Marius M Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2006.
- [Desrochers *et al.*, 1992] Martin Desrochers, Jacques Desrosiers, and Marius Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354, 1992.
- [Dingil, 2025] Ali Enes Dingil. Fostering inclusive urban transportation in planning and policy-making: An umbrella review using alarm methodology. *Sustainable Futures*, 9:100420, 2025.
- [Domingo *et al.*, 2021] Darío Domingo, Gaëtan Palka, and Anna M Hersperger. Effect of zoning plans on urban land-use change: A multi-scenario simulation for supporting sustainable urban growth. *Sustainable Cities and Society*, 69:102833, 2021.
- [Du Merle *et al.*, 1999] Olivier Du Merle, Daniel Wileneuve, Jacques Desrosiers, and Pierre Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1-3):229–237, 1999.
- [Gilmore and Gomory, 1961] Paul C Gilmore and Ralph E Gomory. A linear programming approach to the cutting-stock problem. *Operations research*, 9(6):849–859, 1961.
- [Hodges, 2010] Tina Hodges. *Public transportation’s role in responding to climate change*. Diane Publishing, 2010.
- [Hu *et al.*, 2025] Hins Hu, Rhea Goswami, Hongyi Jiang, and Samitha Samaranayake. Optimal micro-transit zoning via clique generation and integer programming. In *2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC)*, pages 583–589. IEEE, 2025.
- [Kooti *et al.*, 2017] Farshad Kooti, Mihajlo Grbovic, Luca Maria Aiello, Nemanja Djuric, Vladan Radosavljevic, and Kristina Lerman. Analyzing uber’s ride-sharing economy. In *Proceedings of the 26th international conference on world wide web companion*, pages 574–582, 2017.
- [Lens, 2022] Michael C Lens. Zoning, land use, and the reproduction of urban inequality. *Annual Review of Sociology*, 48:421–439, 2022.
- [Liu and Ouyang, 2021] Yining Liu and Yanfeng Ouyang. Mobility service design via joint optimization of transit networks and demand-responsive services. *Transportation Research Part B: Methodological*, 151:22–41, 2021.

- [Miller *et al.*, 2016] Patrick Miller, Alexandre G de Barros, Lina Kattan, and SC Wirasinghe. Public transportation and sustainability: A review. *KSCE Journal of Civil Engineering*, 20(3):1076–1083, 2016.
- [Riley *et al.*, 2019] Connor Riley, Antoine Legrain, and Pascal Van Hentenryck. Column generation for real-time ride-sharing operations. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 472–487. Springer, 2019.
- [Sen *et al.*, 2025] Rishav Sen, Abhishek Dubey, Ayan Mukhopadhyay, Samitha Samaranayake, and Aron Laszka. Moveod: Synthesizing origin-destination commute distribution from us census data. *arXiv preprint arXiv:2510.18858*, 2025.
- [U.S. Census Bureau, 2026] U.S. Census Bureau. American community survey 5-year data (2009–2024), 2026. Accessed: 2026-06-02.
- [Van Den Akker *et al.*, 1999] Janna M Van Den Akker, Jan A Hoogeveen, and Steef L van de Velde. Parallel machine scheduling by column generation. *Operations research*, 47(6):862–872, 1999.
- [Vance *et al.*, 1997] Pamela H Vance, Cynthia Barnhart, Ellis L Johnson, and George L Nemhauser. Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, 45(2):188–200, 1997.