Transportation Research Part C 180 (2025) 105341

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Dynamic partitioning of heterogeneously loaded road networks: A two-level regionalization scheme with Monte Carlo tree search

Cheng Hu [a], Jinjun Tang [a,*], Junjie Hu [a], Yaopeng Wang [a], Zhitao Li [a], Jie Zeng [b], Chunyang Han [c]

[a] Smart Transport Key Laboratory of Hunan Province, School of Transport and Transportation Engineering, Central South University, Changsha 410075, China
[b] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China
[c] Faculty of Transportation Engineering, Kunming University of Science and Technology, Kunming 650500, China

ARTICLE INFO

Keywords:
Dynamic partitioning
Macroscopic fundamental diagram
Regionalization
Monte Carlo tree search

ABSTRACT

This paper proposes a novel dynamic road network partitioning framework tailored for hierarchical network control based on macroscopic fundamental diagrams. The framework establishes a subregion-region system that can be used for both dynamic road network partitioning and perimeter control strategies through a two-level regionalization model. The first level of regionalization is formulated as a mixed-integer quadratic programming (MIQP) problem, and a specialized max-p region algorithm is designed to solve it. An adaptive large neighborhood search (ALNS) algorithm is introduced to optimize the road network partitioning at the subregion level. Treating each subregion as a fundamental geographic unit, the second level of regionalization is modeled as a mixed-integer linear programming (MILP) model. Due to the significant reduction in the problem size, this model can be solved exactly using a solver. Subsequently, dynamic road network partitioning is achieved by performing multiple boundary subregion movements at discrete time points, based on past network partitioning solutions. This partitioning update process is described using a Markov decision process (MDP), and a Monte Carlo tree search (MCTS) algorithm is designed to iteratively determine the optimal movement actions. The performance of the two-level regionalization method in static road network partitioning is analyzed using the urban road network of Yuelu District in Changsha, China. The dynamic road network partitioning method is tested through simulations on a grid network and the urban road network of Bilbao, Spain. The results validate the effectiveness of the proposed framework, which provides valuable insights and practical support for embedding dynamic road network partitioning methods into network-level traffic control strategies.

1. Introduction

The network macroscopic fundamental diagram (NMFD or MFD) has garnered increasing attention due to its simplicity and elegance in characterizing the relationship between macroscopic traffic parameters, such as accumulation and travel production (Daganzo, 2007; Geroliminis and Daganzo, 2008; Johari et al., 2021). MFD-based network-level traffic control strategies have been

[*] Corresponding author.
E-mail address: jinjuntang@csu.edu.cn (J. Tang).

https://doi.org/10.1016/j.trc.2025.105341
Received 7 April 2025; Received in revised form 1 September 2025; Accepted 2 September 2025

proposed to address the limitations of conventional link-level control methods (Geroliminis et al., 2012; Haddad et al., 2013; Keyvan-Ekbatani et al., 2015; Ramezani et al., 2015). These strategies rely on well-defined MFDs to describe the macroscopic traffic state within a region (Aalipour et al., 2018; Mariotte et al., 2020; Paipuri et al., 2020; Saffari et al., 2023). However, numerous studies have revealed that the empirical MFDs observed in real-world networks often exhibit undesirable phenomena, such as scatter (Buisson and Ladier, 2009), hysteresis loops (Mahmassani et al., 2013), and bifurcation (Shim et al., 2019), which are primarily caused by the heterogeneity in congestion distribution across the network. The deterioration in MFDs resulting from network heterogeneity is likely to render control strategies both unreliable and impractical (Jiang and Keyvan-Ekbatani, 2023). Previous studies have demonstrated that a plausible solution is to track congestion pockets within the road network and partition the network into multiple regions that are approximately homogeneous in terms of congestion (Lopez et al. 2017a; Ambühl et al., 2019; Johari et al., 2021). Therefore, effective road network partitioning techniques are essential means to facilitate the practical implementations of macroscopic traffic control strategies.

Most existing studies have focused on fixed road network partitioning schemes (Ji and Geroliminis, 2012; Saeedmanesh and Geroliminis, 2017; Gu and Saberi 2019; Saedi et al., 2020). Perimeter control methods built upon these fixed partitioning schemes to achieve traffic flow rebalancing between regions by regulating the transfer flow at constant boundaries (Aboudolas and Geroliminis, 2013; Fu et al., 2021; Tsitsokas et al., 2023; Li et al., 2024; Liu and Gayah, 2024). However, research has demonstrated that initially homogeneous regions can transition to heterogeneous regions over time due to imposed transfer flow constraints and asymmetric demand load (Ding et al., 2022; Jiang and Keyvan-Ekbatani, 2023). Ramezani et al. (2015) addressed the dynamic heterogeneity in their modeling and developed a hierarchical control scheme to improve traffic flow in the network. Nonetheless, fixed boundary perimeter control methods can still struggle to account for the real-time dynamics of traffic flow and demand. As a result, recent perimeter control literature has shifted toward incorporating real-time network partitioning schemes into the control process (Li et al., 2021; Ding et al., 2022; Hamedmoghadam et al., 2022; Jiang and Keyvan-Ekbatani, 2023). Ding et al. (2022) and Hamedmoghadam et al. (2022) have each proposed methods for defining variable buffer zones or potential expansion areas between protected regions and their peripheries. However, these approaches are non-trivial in estimating or assessing the ranges of congested areas and necessitate resolving the challenge of real-time MFD estimation (Ding et al., 2025). A more straightforward and effective class of approaches builds upon hierarchical network modeling, as introduced in Ramezani et al. (2015). These studies start with a pre-established “region-subregion” multi-reservoir (or multi-cell) system, where the initial partitioning is dynamically updated using simple methods to adapt to the evolving congestion distribution in the network (Li et al., 2021; Jiang and Keyvan-Ekbatani, 2023). However, methods for constructing such hierarchical network structures on real road networks and general approaches for dynamically updating network partitions have yet to be thoroughly explored. In contrast, dynamic network partitioning studies leverage empirical data to track the dynamic patterns of traffic flow on real road networks and explore methods for the evolution of network partitioning (Lopez et al. 2017a; Lopez et al. 2017b; Saeedmanesh and Geroliminis, 2017; Bellocchi and Geroliminis, 2020; Yan et al. 2021). However, these methods are often incompatible with perimeter control strategies. The underlying reasons are twofold. First, regardless of whether the MFD is considered an inherent physical property of the network or a traffic performance curve derived from observed data, any modification to the network structure necessitates a re-estimation of the MFD. As a result, defining a reliable MFD curve for control purposes within a region with dynamically changing boundaries is inherently challenging. Second, the role of network partitioning in perimeter control essentially involves identifying the participants of the MFD-based coordination mechanism. In other words, the established boundaries determine all traffic signals that require coordination, and this coordination mechanism must be consistent with the traffic dynamics described by the regional MFD. Dynamic network partitioning methods achieve real-time adjustments by reallocating road segments and merging or dividing regions. In essence, these methods design a group-based coordination mechanism that does not rely on the regional MFD, making them more suitable for analyzing multi-intersection adaptive signal coordination, as demonstrated in the study by Ma and Wu (2023). Thus, there remains a gap between the theoretical analysis and practical application of real-time road network partitioning that needs to be closed.

This study fills the gap by proposing a dynamic road network partitioning framework tailored to network-level control strategies. A two-level regionalization approach is proposed to address the static road network partitioning problem. In the first level, the regionalization process is formulated as a mixed-integer quadratic programming (MIQP) model. A max-p region algorithm is devised to solve the MIQP model and generate as many compact and approximately homogeneous subregions as possible. A customized adaptive large neighborhood search (ALNS) algorithm is introduced to efficiently optimize the subregion partitioning. The ALNS metaheuristic was initially developed by Ropke and Pisinger (2006a), Ropke and Pisinger (2006b), and Pisinger and Ropke (2007) as an enhancement to the large neighborhood search (LNS) algorithm (Shaw, 1998). Similar to LNS, ALNS employs a destroy operator to disrupt the structure of an initial solution, followed by a repair operator to restore the disrupted solution and generate a neighborhood solution. However, ALNS offers greater flexibility by allowing the modeler to develop a set of problem-specific destroy and repair operators. Moreover, the algorithm adaptively selects combinations of these operators during the iterative process to improve the current solution. These generated subregions are then taken as the decision units in the second level, where a coarser regionalization process is applied to form the final regions. By linearizing both the objective function and constraints, the regionalization problem in the second level is formulated as a mixed-integer linear programming (MILP) model. The model is then solved using an optimization solver to obtain the optimal regional partition. The substantial reduction in the number of decision units significantly prunes the search space of the MILP. While the problem remains NP-hard in general, this reformulation enables exact algorithms in state-of-the-art solvers to consistently find optimal solutions within practical time limits for real-world network sizes. This two-level formulation is designed to align with perimeter control methods based on the “region-subregion” dual-scale framework. Subsequently, a dynamic response mechanism based on the Monte Carlo Tree Search (MCTS) algorithm is employed to enable real-time network partitioning. This mechanism monitors the heterogeneity within each region and iteratively adjusts the boundaries of regions exhibiting deteriorating homogeneity. Notably, the new partitioning is achieved not by modifying individual boundary links, but by adjusting multiple subregions at the boundaries to maintain rough congestion homogeneity within the regions. This approach significantly improves the efficiency of dynamic partitioning and is compatible with theoretical perimeter control frameworks. The proposed methods are validated on both real and simulated road networks.

1.1. Related works

In this section, we primarily review road network partitioning methods and briefly revisit macroscopic traffic flow dynamics modeling and control methods based on MFDs and multi-reservoir systems.

The concept of network-level macroscopic traffic models originated from the studies of Godfrey (1969), Herman and Prigogine (1979), Mahmassani et al. (1984), and Mahmassani et al. (1987). Based on field data, Geroliminis and Daganzo (2008) empirically confirmed the existence of the MFD on the road network in Yokohama, Japan. Subsequently, researchers have employed a range of data sources, including loop detector data (LDD) (Buisson and Ladier, 2009; Ambühl et al., 2021), floating car data (FCD) (Saffari et al., 2023), a combination of LDD and FCD (Ambühl and Menendez, 2016; Du et al., 2016; Zockaie et al., 2018; Saffari et al., 2020), mobile phone data (Paipuri et al., 2020), and automatic license plate recognition data (ALPR) (Niu et al., 2022; Hu et al., 2025a), whether collected or simulated, to estimate the MFDs in real road networks. Due to the sparse distribution of detectors, several studies have focused on identifying critical links and optimizing sensor allocation to improve MFD estimation (Zockaie et al., 2018; El Bukhari et al., 2024; Mousavizadeh and Keyvan-Ekbatani, 2024). In the early works, single-reservoir MFD models and their associated control methods were designed specifically for small road networks characterized by negligible demand fluctuations and roughly homogeneous congestion distribution (Daganzo, 2007; Geroliminis and Daganzo, 2008; Daganzo and Geroliminis, 2008). These stringent conditions are rarely met in real road networks, given the uneven demand distribution and evolving nature of congestions. As a result, a growing body of research have moved toward multi-reservoir traffic flow modeling, such as route choice behavior (Leclercq et al., 2015; Yildirimoglu et al., 2015), dynamic equilibrium analysis (Yildirimoglu and Geroliminis, 2014; Huang et al., 2020; Batista et al., 2025; Duncan et al., 2025), dynamic traffic assignment (Batista and Leclercq, 2019; Batista et al., 2021), network loading (Mariotte and Leclercq, 2019; Moshahedi and Kattan, 2022), and model calibration and validation (Paipuri et al., 2019; Mariotte et al., 2020; Paipuri et al., 2020), as well as multi-reservoir control strategies, including perimeter control (Aboudolas and Geroliminis, 2013; Ramezani et al, 2015; Haddad and Zheng, 2020; Yu et al., 2025), perimeter control combined with route guidance (Sirmatel and Geroliminis, 2017; Yildirimoglu et al., 2018; Chen et al., 2024a), perimeter control considering boundary queuing (Fu et al., 2021; Keyvan-Ekbatani et al., 2021), and pricing (Zheng and Geroliminis, 2016; Genser and Kouvelas, 2022; Loder et al., 2022). As a fundamental building block, the road network must be partitioned into multiple homogeneous, compact, and connected regions (multi-reservoir systems), with the assumption that each region is associated with a well-defined MFD.

To facilitate the implementation of the macroscopic control methods on heterogeneous networks, numerous studies have focused on utilizing link traffic flow data while taking into account the network topological structure to partition real-world road networks. The pioneering study proposed an “initial segment-merge-boundary adjustment” framework to obtain multiple nearly homogeneous and compact subregions (Ji and Geroliminis, 2012), building on the normalized cut algorithm (Shi and Malik, 2000). The method works well on regular networks, but cannot handle asymmetric network topologies and highly directional traffic flows. To capture the significant spatial correlation of congestion propagation, Saeedmanesh and Geroliminis (2016) designed a “snake” algorithm to establishes similarity relations between disconnected nodes and used the Symmetric Nonnegative Matrix Factorization (SNMF) to conduct road network partitioning. The remaining studies also utilized clustering methods that focus solely on the characteristics of the data points themselves, such as K-means (Lopez et al., 2017a) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Lopez et al. 2017a; Lopez et al. 2017b), along with spectral clustering algorithms (Ji et al., 2014; Pascale et al., 2015; Gu and Saberi, 2019; Saedi et al., 2020; Yan et al., 2021), MILP models (Saeedmanesh and Geroliminis, 2017), and graph neural networks (Jiang et al., 2023; Hu et al., 2025b) for partitioning heterogeneously loaded road networks (see Hu et al., 2025b for a comprehensive methodological review).

It is worth noting that most studies have focused solely on static road network partitioning (Zhang et al., 2020). In the literature, dynamic road network partitioning involves continuously fine-tuning the initial partitioning based on incoming link data using a predefined mechanism. For instance, Saeedmanesh and Geroliminis (2017) and Yan et al. (2021) each developed a dynamic update mechanism based on the “expansion-shrinking-merging-splitting” approach. These studies place more emphasis on tracking homogeneous congestion pockets in both spatial and temporal dimensions, while also allowing for dynamic changes in the number of clusters. However, non-compact road network partitions and their frequent alterations can complicate macroscopic dynamic modeling and conflict with the requirements of network-level traffic flow control methods. Several studies on perimeter control have introduced distinctive dynamic boundary approaches. For example, to mitigate queue formation and congestion spillover caused by conventional perimeter control, Hamedmoghadam et al. (2022) proposed a dynamic boundary identification method based on percolation theory to regulate congestion propagation in real time. Similarly, Ding et al. (2022) examined the potential expansion of central congested zones to address the outward spread of congestion and developed a three-dimensional MFD surface model to characterize the influence of regional boundary changes on the MFD. However, a set of discrete MFD curves cannot fully capture variations in regional shape, making the calibration of such a model challenging in real-world networks. To address the calibration challenges, an alternative stream of research has focused on model-free perimeter control approaches that relax the dependence on model calibration (Zhou and Gayah, 2021; Zhou and Gayah, 2023; Hu and Ma, 2024). These methods typically employ reinforcement learning or other data-driven techniques to directly optimize control actions without requiring any macroscopic traffic flow dynamics information (Chen et al., 2022; Chen et al., 2024b). However, most model-free approaches assume fixed regional boundaries and have rarely considered

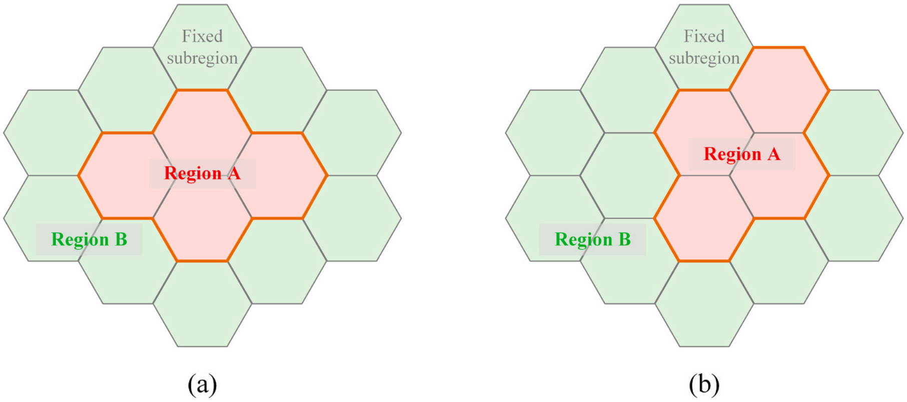
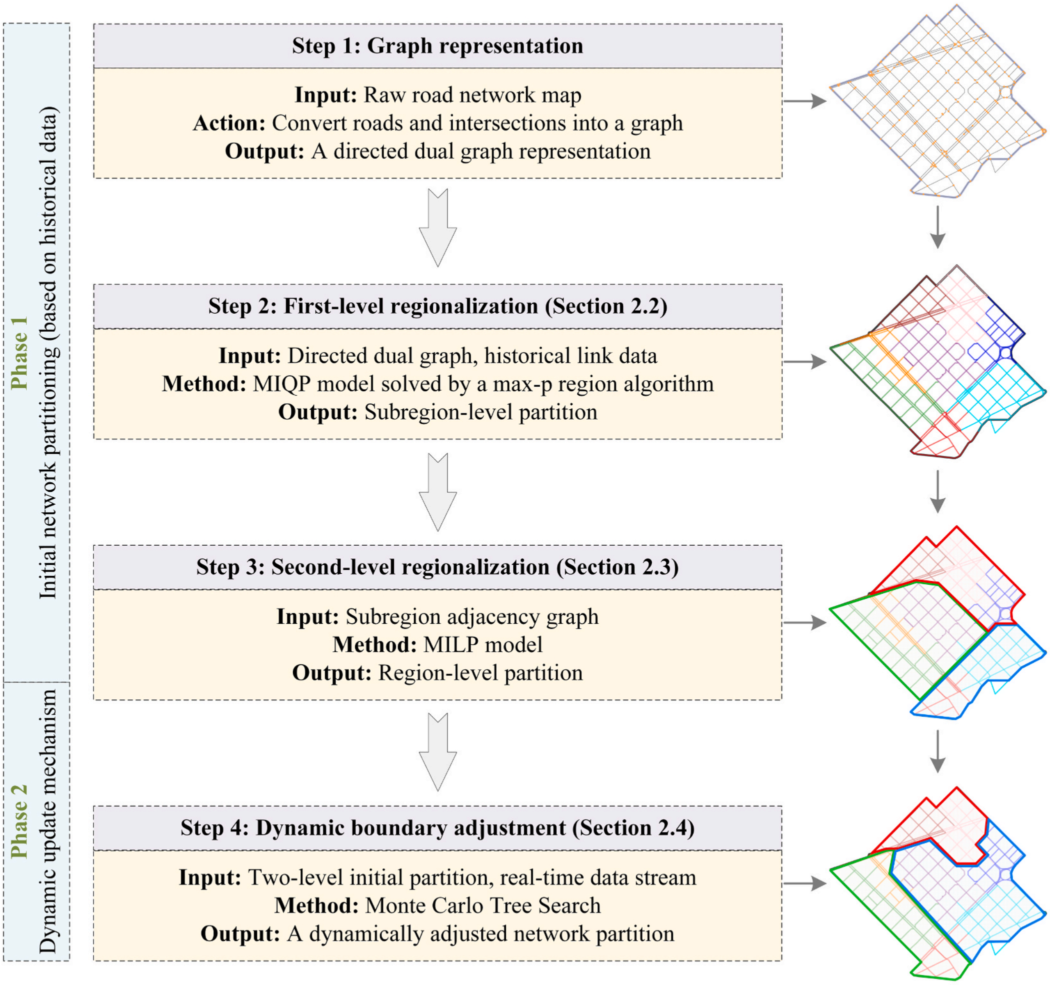


Fig. 1. (a) A network with two regions and fourteen subregions; (b) evolution of network partitioning at the region level.



Step 1: Graph representation

Input: Raw road network map
Action: Convert roads and intersections into a graph
Output: A directed dual graph representation

Step 2: First-level regionalization (Section 2.2)

Input: Directed dual graph, historical link data
Method: MIQP model solved by a max-p region algorithm
Output: Subregion-level partition

Step 3: Second-level regionalization (Section 2.3)

Input: Subregion adjacency graph
Method: MILP model
Output: Region-level partition

Step 4: Dynamic boundary adjustment (Section 2.4)

Input: Two-level initial partition, real-time data stream
Method: Monte Carlo Tree Search
Output: A dynamically adjusted network partition

Fig. 2. Methodological framework.

dynamic partitioning strategies. Another line of research integrates real-time road network partitioning into perimeter control by employing a multi-reservoir system composed of compact subregions and regions (as depicted in Fig. 1(a)) (Li et al., 2021; Jiang and Keyvan-Ekbatani, 2023). The partitioning at the subregion level remains fixed, while the partitioning at the region level is adjusted in response to the evolving congestion patterns (as demonstrated in Fig. 1(b)). This approach effectively circumvents the need for re-

Table 1
Description of notations.

Notation	Description
Graph representation	
$G_o = (V_o, E_o)$	Original directed graph with a node set V_o and an edge set E_o
$G = (V, E)$	Transformed dual directed graph with a node set V and an edge set E
$G_s = (\mathscr{V}, \mathscr{E})$	Subregion adjacency graph with a node set $\mathscr{V}$ and an edge set $\mathscr{E}$
Regionalization model	
R	Set of subregions
C	Set of hierarchical order
$x^{i,r,c}$	Binary decision variable that equals 1 when node $i \in V$ is assigned to subregion $r \in R$ in a hierarchical order of $c \in C$
$z^{i,j,r}$	Binary decision variable that equals 1 when both nodes $i \in V$ and $j \in V$ belong to subregion $r \in R$
s^i	Average congestion level of link(node) $i \in V$
$\bar{s}^r$	Average congestion level of subregion $r \in R$
$\bar{s}^V$	Average congestion level of the entire network
θ	Minimum threshold for the number of nodes within each subregion
λ_1, λ_2	Trade-off coefficients in the first-level regionalization model
$\mathscr{R}$	Set of regions
$x_{i,r}$	Binary decision variable that equals 1 when node (i.e., subregion) $i \in \mathscr{V}$ is assigned to region $r \in \mathscr{R}$
$\mathscr{z}_{i,j,r}$	Binary decision variable that equals 1 when the edge $(i,j) \in \mathscr{E}$ belongs to region $r \in \mathscr{R}$
$k_{i,r}$	Binary decision variable that equals 1 when node $i \in \mathscr{V}$ is the root node of region $r \in \mathscr{R}$
$f_{i,j,r}$	Decision variable that represents the flow from node $i \in \mathscr{V}$ to node $j \in \mathscr{V}$ within region $r \in \mathscr{R}$
$\mathscr{D}$	Cumulative congestion level differences on all connected edges in the graph
s_i	Average congestion level within subregion $i \in \mathscr{V}$
η	Minimum threshold for the number of subregions within each region
n_r	Number of subregions in region $r \in \mathscr{R}$
$y_{i,r}$	Auxiliary decision variable used for the linearization
λ_3, λ_4	Trade-off coefficients in the second-level regionalization model
Max-p region algorithm	
Φ^-	Set of destroy operators
Φ^+	Set of repair operators
A	Optional matrix that contains all actions (destroy-repair operator pairs)
$T_a(t-1)$	Number of times action a has been executed
$\bar{\pi}_a(t)$	Average reward of action a in the first t iterations
α	Hyperparameter that regulates the intensity of exploration and controls the influence of the exploration bonus on the value of an action
Monte Carlo tree search algorithm	
$\mathscr{S}$	Complete configuration of the road network system at any decision time point
$\mathscr{K}$	Set of current regional mean densities
$\mathscr{B}$	Set of boundary subregion indices
$\mathscr{H}$	Set of recent movement histories for all subregions
$\mathscr{A}(\mathscr{s})$	Action space given state $\mathscr{s} \in \mathscr{S}$
$\mathscr{b}$	Boundary subregion to be moved
$N(\mathscr{b}, \mathscr{R})$	Set of adjacent regions of subregion $\mathscr{b} \in \mathscr{B}$
$\mathscr{P}$	State transition function
$\Lambda(\mathscr{s}, \mathscr{a}, \tilde{\mathscr{s}})$	Reward function that computes the immediate reward of taking action $\mathscr{a}$ and reaching state $\tilde{\mathscr{s}}$
$\Delta CV(\mathscr{s}, \tilde{\mathscr{s}})$	Changes in the coefficient of variation between two states
$\Delta NS(\mathscr{s}, \tilde{\mathscr{s}})$	Changes in the average Ncut Silhouette between two states
$I_{V\text{Constraint penalty}}(\tilde{\mathscr{s}})$	Penalty for violating the region connectivity and size constraints
γ	Tuning parameter that governs the preference for different objectives at each phase of the process
$\mathscr{Q}(\mathscr{s}, \mathscr{a})$	Average rewards of performing action $\mathscr{a}$ in state $\mathscr{s}$ in the simulation
$\mathscr{N}(\mathscr{s})$	Number of simulation times
$\mathscr{N}(\mathscr{s}, \mathscr{a})$	Number of times action $\mathscr{a}$ was performed
ω	Trade-off parameter that controls preference between exploration and exploitation

estimating regional MFDs while offering a compromised solution for managing boundary queues and intra-regional heterogeneity. Moreover, it preserves the elegance of macroscopic traffic flow modeling and maintains the computational efficiency of dynamic partitioning. However, existing studies have not adequately addressed the challenge of developing multi-scale partitioning models for real-world road networks or a general, effective approach to updating dynamic network partitioning. Therefore, this study aims to fill this gap and pave the way for the practical application of the methods proposed by Li et al. (2021) and Jiang and Keyvan-Ekbatani (2023).

1.2. Contributions and organization

The contributions of this study are summarized as follows.

- A novel framework for dynamic network partitioning tailored to hierarchical network control strategies is proposed. The framework enables the construction of a subregion-region system on real road networks and introduces a general dynamic updating approach to adapt region-level partitions to the evolving patterns of traffic congestion.

- A two-level regionalization model with distinct objectives is developed to construct the subregion-region system. The first-level regionalization is formulated as a MIQP problem, with the objective of generating as many homogeneous and compact subregions as possible. A meticulously designed max-p region algorithm is proposed to solve this MIQP problem while maintaining computational efficiency. The second-level regionalization is formulated as a small-scale MILP problem to identify well-differentiated regional boundaries.
- A dynamic updating mechanism based on the MCTS algorithm is developed to optimize region-level network partitions in real time. The anytime property of the MCTS algorithm ensure that the current best solution can be obtained within a limited computation time. Operating at the subregion level significantly reduces the problem scale, thereby mitigating the computational burden of the MCTS algorithm and facilitating its application to large-scale road networks.
- Case studies are conducted on two real road networks and one simulated grid network to comprehensively validate the effectiveness of the proposed framework. The results indicate that smaller region sizes are more conducive to maintaining intra-regional homogeneity over time. Moreover, moderately longer network partitioning update intervals are more effective in optimizing the performance of dynamic partitioning.

The remainder of the study is organized as follows. Section 2 introduces the dynamic network partitioning framework. Section 3 presents the static partitioning case study results. Section 4 presents the case study results on dynamic partitioning using a simulated grid network. Section 5 further validates the framework through experiments on a real urban road network. Section 6 provides some concluding remarks.

2. Dynamic network partitioning framework

This section primarily presents the proposed framework for dynamic road network partitioning, as illustrated in Fig. 2. The framework consists of two phases. In the first phase, an initial two-level partitioning of the road network, comprising subregions and regions,is derived from historical traffic data. The road network is represented as a directed dual graph. The partitioning problem at the subregion level is formulated as a MIQP problem, and a max-p region algorithm is developed to solve the model. Subsequently, a subregion adjacency graph is constructed, and the regionalization of this graph is expressed as a MILP problem. Solving this problem yields the region-level road network partitioning scheme. In the second phase, an MCTS-based dynamic update mechanism is introduced to continuously refine the regional boundaries in response to evolving real-time traffic congestion patterns. The notations used in this section are summarized in Table 1.

2.1. Problem descriptions

The road network is represented as a directed graph $G_o = (V_o, E_o)$, where V_o and E_o denote the sets of nodes and edges, respectively. For the purpose of network partitioning, links are selected as the partitioning entities. Consequently, the graph is transformed into an undirected dual graph $G = (V, E)$, where the node set V represents the links, and the edge set E reflects the connectivity between these links. It is generally assumed that congestion propagates through the road network at a finite speed. Therefore, the changes in the road network partitioning scheme over time are relatively slow. We assume that updates to the partitioning scheme occur at discrete time points, separated by fixed intervals, with each pair of consecutive time points defining a control horizon. The objective of dynamic road network partitioning is to identify a series of temporally discrete, interdependent partitioning schemes. In other words, each road network partitioning scheme at a given time point is derived from traffic flow data within the previous control horizon, along with historical partitioning schemes. At the beginning of the time frame, a static road network partitioning scheme, derived from historical data (such as average congestion measures), is employed to initiate the dynamic update process.

To establish a system composed of subregions and their corresponding regions, we employ hierarchical regionalization at different scales to partition the road network. Regionalization, a concept from Geographic Information Systems (GIS) field, aims to aggregate adjacent geographical units into clusters that satisfy certain internal criteria (Assunção et al., 2007; Guo, 2008; Kirkley, 2022). In this context, initial regionalization treats each roads/links (i.e., nodes on the graph G) as a basic geographical unit and aggregates them into smaller, connected, compact, and approximately homogeneous subregions. At a larger scale, regionalization treats each subregion as a geographical unit, and neighboring subregions collectively form regions. The rationale for adopting this approach, rather than a direct duplication of conventional road network partitioning methods, is that partitioning at different scales serves distinct objectives and requirements. Specifically, the first level of regionalization or road network partitioning aims to divide the road network into pockets with varying congestion levels. These pockets reflect long-term, strong correlations in local traffic flows and are therefore robust in the time dimension. Theoretically, if the granularity is sufficiently fine, the road network can always be partitioned into compact and homogeneous units. Furthermore, regionalization at a small scale should not dictate the total number of subregions. Since these subregions are fixed in subsequent models, the model should inherently possess the capability to determine the optimal number of subregions. In perimeter control studies, the primary purpose of defining higher level partitions at the subregion level is to reduce the boundaries (e.g., intersection signals) that need to be controlled, thereby enhancing control efficiency. For multi-reservoir systems, the length of controlled boundaries increases significantly with the number of reservoirs. However, not every boundary (i.e., between two reservoirs) requires attention. This level of regionalization aims to identify boundaries between subregions where congestion levels increase or decrease significantly, with the aim of restricting transfer flows at these boundaries by controlling traffic signals. Typically, boundaries at the regional level do not change drastically in a short period of time. Consequently, it is assumed that the number of regions is fixed and remains constant during the modeling process, as splitting or merging regions would lead to significant changes in control boundaries, resulting in substantial fluctuations in control. However, continuous minor adjustments over time can lead to new partitioning schemes that clearly distinguish themselves from the initial partitioning.

2.2. First-level regionalization: subregion generation

This section derives the road network partitioning scheme at the subregion level by solving a regionalization problem. The regionalization problem is formulated as a max-p region model, which endogenously determines the number of subregions (Duque et al., 2012; She et al., 2016). The objective of this model is to maximize the number of connected, compact, and homogeneous subregions. To achieve this, we utilize normalized total variance (TV_n) (Ji and Geroliminis, 2012) as the homogeneity measure and the boundary edge ratio as the compactness measure. Intuitively, a more compact network partitioning scheme should have fewer boundary edges. These two objectives are weighted and incorporated into the optimization function. Furthermore, a minimum threshold for the number of nodes (or links) within a subregion is defined to ensure that the resulting subregions are balanced in size. Thus, the first-level regionalization can be formally expressed as the following MIQP problem:

$$min\left(-\sum_{r\in R}\sum_{i\in V}x^{i,r,0}\right)\cdot 10^F+\lambda_1\cdot\frac{\sum_{r\in R}\left[\sum_{i\in V}\left(\sum_{c\in C}x^{i,r,c}\right)\cdot\left(s^i-\bar{s}^r\right)^2\right]}{\sum_{i\in V}\left(s^i-\bar{s}^V\right)^2}+\lambda_2\cdot\frac{\sum_{(i,j)\in E}\left(1-\sum_{r\in R}z^{i,j,r}\right)}{|E|},\tag{1}$$

Subject to:

$$F=1+\left[\log\left(\sum_{i\in V}\sum_{r\in R}\sum_{c\in C}x^{i,r,c}\right)\right],\tag{2}$$

$$\bar{s}^r=\frac{\sum_{i\in V}\left(s^i\cdot x^{i,r,c}\right)}{\sum_{i\in V}x^{i,r,c}},\quad\forall r\in R,\ c\in C,\tag{3}$$

$$\bar{s}^V=\frac{\sum_{i\in V}s^i}{|V|},\tag{4}$$

$$\sum_{r\in R}\sum_{c\in C}x^{i,r,c}=1,\quad\forall i\in V,\tag{5}$$

$$\sum_{i\in V}x^{i,r,0}=1,\quad\forall r\in R,\tag{6}$$

$$x^{i,r,c}\le\sum_j x^{j,r,c-1},\quad\forall(i,j)\in E,\ r\in R,\ c\in C,\tag{7}$$

$$\sum_{i\in V}\sum_{c\in C}x^{i,r,c}\ge\theta\sum_{i\in V}x^{i,r,0},\quad\forall r\in R,\tag{8}$$

$$z^{i,j,r}\le\sum_{c\in C}x^{i,r,c},\quad z^{i,j,r}\le\sum_{c\in C}x^{j,r,c},\quad z^{i,j,r}\ge\sum_{c\in C}x^{i,r,c}+\sum_{c\in C}x^{j,r,c}-1,\quad\forall(i,j)\in E,\ r\in R,\tag{9}$$

$$x^{i,r,c}=\begin{cases}1, & \textit{if node } i \textit{ is assigned to subregion } r \textit{ in order } c,\\ 0, & \textit{otherwise}.\end{cases}\tag{10}$$

In the objective function, the first term guides the model to generate as many subregions as possible that satisfy the internal threshold, i.e., the minimum number of nodes required within each subregion, by applying a large penalty to a lower number of subregions. The second term calculates the normalized total variance, while the third term computes the boundary edge ratio. λ_1 and λ_2 are two trade-off weights. Constraint (9) indicates that the decision variable $z^{i,j,r}$ equals 1 only when both nodes i and j belong to subregion r; otherwise, it equals 0. Constraint (10) indicates that the decision variable $x^{i,r,c}$ equals 1 when node i is assigned to subregion r in a hierarchical order of c; otherwise, it equals 0. R and C are the set of subregions and hierarchical orders, respectively. Note that the model defines a root node ($c = 0$) for each subregion, with the multi-hop distance between other nodes and the root node reflecting their hierarchical orders. Therefore, the decision variable definitions inherently ensure connectivity within each subregion. Constraint (2) provides the calculation of the exponential term in the penalty function. Constraints (3) and (4) define the average congestion level (i.e., speed or density) for each subregion and the entire road network, respectively, where s^i, $\bar{s}^r$ and $\bar{s}^V$ represent the average congestion levels of link i, subregion r, and the entire network, respectively. Constraint (5) ensures that each node is assigned to only one subregion at a specific hierarchical order. Constraint (6) ensures that each subregion has exactly one root node. Constraint (7) ensures that each node in a subregion has at least one neighboring node within the same subregion with a higher hierarchical order. Constraint (8) ensures that the number of nodes within each subregion meets the minimum threshold θ.

It is evident that the above problem, analogous to the original max-p region problem, is NP-hard and can only be solved with exact algorithms for very small road network sizes (Duque et al., 2012). As a result, this section proposes a three-phase heuristic approach to solve the model. This approach draws inspiration from the “region growth-enclave assignment-local search” framework presented by Wei et al. (2020). Briefly, during the region growth and enclave assignment phases, an initial subregion-level road network partitioning solution is generated by constructing a BFS tree around randomly selected seed nodes to satisfy the node count threshold, and by allocating the remaining nodes using a randomized greedy strategy. The maximum number of subregions is determined by executing the region growth and enclave assignment phases multiple times and selecting the solution that results in the highest number of subregions. A detailed description of the first two phases can be found in Wei et al. (2020) and Hu et al. (2025a). The local search phase then further refines the partitioning solution, with a fixed number of subregions, to minimize the second and third terms in the objective function, specifically the homogeneity and compactness within the subregions. Previous studies have utilized simulated annealing, tabu search, and their combinations as refinement algorithms for this phase (Duque et al., 2012; Wei et al., 2020). These methods explore the neighborhood by moving one node at a time, which is sufficiently efficient for regionalization problems, as they typically involve a relatively small number of geographical units. However, road network partitioning problems often involve thousands of nodes, making these approaches inefficient due to their slow neighborhood exploration. To address this issue, we adopt an ALNS algorithm to improve the initial road network partitioning solution. This method has been proven effective in large-scale combinatorial optimization problems, such as the vehicle routing problem and its variants (Mara et al., 2022). Herein, we focus on presenting the components of the customized ALNS algorithm and highlighting its differences from previous approaches.

The design of the operators and the adaptive selection mechanism constitute two fundamental components of ALNS algorithm. To improve the homogeneity and compactness of the generated subregions, we have meticulously developed four destroy operators and five repair operators, which are detailed as follows:

Boundary destroy operator: For each subregion, this operator identifies boundary nodes that are connected to nodes outside the subregion and randomly removes a certain proportion of these nodes.

Random destroy operator: For each subregion, this operator randomly removes a specified proportion of nodes (excluding the root node) while ensuring that the subregion remains connected after each removal.

Greedy destroy operator: For each subregion, this operator sorts the nodes based on the difference in their congestion level (i.e., speed or density) with the subregion, from largest to smallest, and sequentially removes a certain proportion of nodes (excluding the root node), while maintaining connectivity within the subregion after each removal.

Hierarchical destroy operator: For each subregion, this operator selects nodes with a hierarchical order greater than a specified threshold with respect to the root node, randomly removes a certain proportion of these nodes, and ensures that the subregion remains connected after each removal.

Greedy repair operator 1: This operator assigns all unallocated nodes to the neighboring subregion with the smallest congestion level difference, relative to the average congestion level of the subregion.

Greedy repair operator 2: For each unallocated node, this operator calculates the number of adjacent nodes in each subregion and assigns the node to the subregion with the highest adjacency count.

Proximity repair operator: This operator prioritizes assigning unallocated nodes to neighboring subregions closest to the root node that have not yet reached the node count threshold. If no such subregions are available, the node is assigned to the closest neighboring subregion, regardless of the node count.

Random repair operator: This operator randomly assigns unallocated nodes to neighboring subregions that have not yet met the node count threshold.

Local adjust repair operator: This operator first assigns unallocated nodes to neighboring subregions that have not yet met the node count threshold. For each node, the operator examines the subregion of its neighbors and moves the node to the subregion with a higher number of neighbors to enhance local connectivity.

It is worth noting that all the destroy operators incorporate a connectivity check for the subregions to prevent the disruption of subregion connectivity. Furthermore, some of the repair operators prioritize subregions that have not yet met the node count threshold, thereby ensuring a balanced subregion size. In each iteration, the algorithm selects a pair of operators from the destroy operator set $\Phi^- = \{\Phi_1^-, \Phi_2^-, \Phi_3^-, \Phi_4^-\}$ and the repair operator set $\Phi^+ = \{\Phi_1^+, \Phi_2^+, \Phi_3^+, \Phi_4^+, \Phi_5^+\}$ to modify the current solution and generate a candidate solution. The standard ALNS algorithm utilizes a roulette wheel method for operator selection and applies the Metropolis criterion to determine whether to accept the candidate solution as the new current solution (Mara et al., 2022). The roulette wheel mechanism assigns weights to operators based on their historical performance, with the selection probability being proportional to these weights. While this approach is computationally efficient, its strong focus on exploitation may result in premature convergence to a local optimum. Accordingly, the present study employs a multi-armed bandit strategy to balance exploration and exploitation, thereby maximizing long-term cumulative rewards and mitigating the risk of premature convergence. Specifically, we adopt a method known as the α-Upper Confidence Bound (α-UCB) approach (Hendel, 2022). This method initializes an optional matrix containing all possible combinations of destroy-repair operator pairs. In each iteration, the exploration reward for each operator pair is calculated and combined with its historical average reward to yield a cumulative value. The operator pair with the highest value is then selected for execution in each iteration t, which can be described as:

$$Q(t) = argmax_{a \in A} \left\{ \overline{\pi}_a(t-1) + \sqrt{\frac{\alpha \cdot \ln(1+t)}{T_a(t-1)}} \right\}, \tag{11}$$

$$\overline{\pi}_a(t) = \frac{\overline{\pi}_a(t-1) \cdot T_a(t-1) + scores[outcome]}{T_a(t-1)+1}, \tag{12}$$

where $Q(t)$ denotes the action selection in iteration t, which determines a pair of destroy-repair operators to be executed; $T_a(t-1)$

Table 2
Scores of execution outcomes.

Outcomes	Scores
A new global best solution.	3
An improved solution (relative to the current solution).	2
An accepted solution with no improvement.	1
A rejected solution.	0

denote the number of times action a has been executed, and A represents the optional matrix that contains all actions (destroy-repair operator pairs). Eq. (11) illustrates that the exploration bonus term increases logarithmically with the global iteration count and is inversely proportional to the number of times an action has been selected, thereby promoting the exploration of actions that have been insufficiently explored. The hyperparameter α regulates the intensity of exploration and controls the influence of the exploration bonus on the value of an action. $\overline{\pi}_a(t-1)$ represents the average reward of action a in the first $t-1$ iterations, and Eq. (12) outlines the reward update rule after an action is executed, where $T_a(t) = T_a(t-1) + 1$. Note that the average rewards at iteration 0 is initialized to 1, i.e., $\overline{\pi}_a(0) = 1$ for $a \in A$. The $scores[outcome]$ term indicates that the score is calculated based on the outcome of executing action a. The scores for the different outcomes are shown in Table 2.

In addition, we use the maximum number of iterations as the termination criterion for the ALNS algorithm. Since the root node in the initial solution is randomly generated, and as the destroy-repair process proceeds, the root node may no longer remain in the center of each subregion. As a result, we update the root node for each subregion at the beginning of the algorithm and after every fixed number of iterations, replacing it with the current (topologically) central node of the respective subregion. The pseudocode of the proposed customized ALNS can be found in the Appendix A.1.

2.3. Second-level regionalization: region identification

This section introduces an additional regionalization process to aggregate the subregions obtained in the previous section into regions, thereby achieving a network partition at the regional level. As previously mentioned, the total number of subregions generated by the map-p region algorithm is actually controlled by the internal threshold θ of the subregions. In any case, this internal threshold should be sufficiently large to ensure the macroscopic modelling significance. This implies that the final number of subregions is significantly reduced in comparison to the number of links in the network, leading to a dramatic reduction in the scale of the regionalization problem. Consequently, this enables the application of exact algorithms to solve the regionalization problem at this level.

However, the quadratic term in the calculation of TV_n in Eq. (1) significantly impairs the efficiency of the model solution process. In addition, the regionalization at this level is primarily concerned with the significant differences of congestion level between adjacent regions. As a result, we replace the homogeneity metric TV_n with the sum of the absolute deviations in congestion level between connected nodes within each subregion. By minimizing this metric, the model is encouraged to generate network partitions with substantial differences at the boundaries. Based on the subregion partitions, a subregion adjacency graph $G_s = (\mathscr{V}, \mathscr{E})$ can be constructed, where $\mathscr{V}$ represents the set of nodes corresponding to the subregions, and $\mathscr{E}$ represents the set of edges reflecting the adjacency relationships between subregions. The regionalization problem on this adjacency graph can then be formulated as the following MILP problem:

$$min\lambda_3 \cdot \frac{\sum_{(i,j)\in\mathscr{E}}\sum_{r\in\mathscr{R}}\left|s_i - s_j\right| \cdot \mathscr{z}_{i,j,r}}{\mathscr{D}} + \lambda_4 \cdot \frac{\sum_{(i,j)\in\varepsilon}\left(1 - \sum_{r\in\mathscr{R}}\mathscr{z}_{i,j,r}\right)}{|\mathscr{E}|}, \tag{13}$$

Subject to:

$$\sum\nolimits_{r\in\mathscr{R}} x_{i,r} = 1, \quad \forall i \in V, \tag{14}$$

$$\sum\nolimits_{i\in\mathscr{V}} x_{i,r} \geq \eta, \quad \forall r \in \mathscr{R}, \tag{15}$$

$$\mathscr{z}_{i,j,r} \leq x_{i,r}, \ \mathscr{z}_{i,j,r} \leq x_{j,r}, \ \mathscr{z}_{i,j,r} \geq x_{i,r} + x_{j,r} - 1, \quad \forall (i,j) \in E, \ r \in R, \tag{16}$$

$$\sum\nolimits_{i\in\mathscr{V}} k_{i,r} = 1, \quad \forall r \in \mathscr{R}, \tag{17}$$

$$k_{i,r} \leq x_{i,r}, \quad \forall i \in V, \ r \in R, \tag{18}$$

$$\sum\nolimits_{(i,j)\in\mathscr{E}} f_{i,j,r} - \sum\nolimits_{(j,i)\in\mathscr{E}} f_{j,i,r} = n_r \cdot k_{i,r} - x_{i,r}, \quad \forall i \in V, \ r \in R, \tag{19}$$

$$f_{i,j,r} \leq M \cdot \mathscr{z}_{i,j,r}. \tag{20}$$

The primary variables are distinguished from the superscripts in the subregion partitioning model by using subscripts. For example, s_i represents the average congestion level within subregion i. The binary decision variable $x_{i,r}$ equals 1 when node (i.e., subregion) i is assigned to region r, and 0 otherwise. The binary decision variable $\mathscr{x}_{i,j,r}$ equals 1 when the edge (i,j) belongs to region r, and 0 otherwise. The binary decision variable $k_{i,r}$ equals 1 when node i is the root node of region r, and 0 otherwise. The decision variable $f_{i,j,r}$ represents the flow from node i to node j within region r. Eq. (13) describes the objective function as a weighted combination of the homogeneity and compactness metrics, where $\mathscr{D}$ represents the congestion level differences on all connected edges in the graph, i.e., $\mathscr{D} = \sum_{(i,j)\in\mathscr{E}}|s_i - s_j|$, $\mathscr{R}$ denotes the set of all regions, and λ_3 and λ_4 are the weighting coefficients. Constraint (14) indicates that each node can only be assigned to one region. Constraint (15) restricts the number of subregions within each region to be greater than or equal to η, i.e., η represents the lower-bound of the number of subregions in a region. Constraint (16) describes the relationship between the membership of edges and the membership of nodes. Constraint (17) ensures that each region contains only one root node. Constraint (18) stipulates that the root node of region r must belong to that region. Constraints (19) and (20) enforce flow conservation for each region to ensure connectivity, where $n_r = \sum_{i\in\mathscr{V}} x_{i,r}$ represents the number of subregions in region r and M is a sufficiently large positive number. Due to the presence of nonlinear terms $(n_r \cdot k_{i,r})$ in Constraint (19), we apply the following linearization procedure:

$$\sum\nolimits_{(i,j)\in\mathscr{E}} f_{i,j,r} - \sum\nolimits_{(j,i)\in\mathscr{E}} f_{j,i,r} = y_{i,r} - x_{i,r}, \quad \forall i \in V,\ r \in R, \tag{21}$$

$$y_{i,r} = n_r \cdot k_{i,r}, \quad \forall i \in V,\ r \in R, \tag{22}$$

$$y_{i,r} \le M \cdot k_{i,r}, \quad \forall i \in V,\ r \in R, \tag{23}$$

$$y_{i,r} \ge 0, \quad \forall i \in V,\ r \in R, \tag{24}$$

$$y_{i,r} \le n_r, \quad \forall i \in V,\ r \in R, \tag{25}$$

$$y_{i,r} \ge n_r - M \cdot \left(1 - k_{i,r}\right), \quad \forall i \in V,\ r \in R, \tag{26}$$

where $y_{i,r}$ is the auxiliary decision variable used for the linearization. Therefore, the above MILP model can be efficiently solved using any open-source or commercial solver to obtain an exact regional-level network partition solution within a relatively short time. The experiment analyzing the computational efficiency of solving the MILP problem using an open-source solver can be found in Appendix A.2. It is important to note that the number of regions must be predefined by the modeler, providing flexibility to adjust the complexity of macroscopic traffic flow modeling and to regulate the length of control boundaries.

2.4. Dynamic update mechanism

As noted above, dynamic road network partitioning is a discrete and interdependent decision-making process. Each decision point involves solving an optimization problem based on newly received traffic flow data and previous partition solutions. Some studies assume that the creation of a new road network solution arises solely from a single subregion movement (decision) within the old network solution (Li et al., 2021). However, this approach is ineffective in identifying appropriate control boundaries when the road network becomes highly heterogeneous. In this section, the road network partitioning optimization problem at discrete time points is treated as a multi-step sequential decision problem, and an MCTS algorithm is designed to solve it.

Given an initial partition solution derived from historical data, the objective is to iteratively optimize the performance of this solution at the current time point by migrating boundary subregions between adjacent regions. Two primary metrics are considered: the coefficient of variation (CV), which measures intra-cluster homogeneity, and the average Ncut Silhouette (NS) (Ji and Geroliminis, 2012), which further assesses inter-cluster heterogeneity. The definitions of these two metrics are provided as follows:

$$CV(r_i) = \frac{\sqrt{Var(r_i)}}{\mu_{r_i}}, \quad \forall r_i \in R, \tag{27}$$

$$NS(r_i) = \frac{2 \cdot Var(r_i)}{Var(r_i) + Var\left(r_j\right) + \left(\mu_{r_i} - \mu_{r_j}\right)^2}, \quad \forall r_i \in R, \tag{28}$$

$$NS = \frac{\sum_{r_i\in\mathscr{R}} NS(r_i)}{|\mathscr{R}|}, \tag{29}$$

where $Var(r_i)$ denotes the variance of subregional space-mean density in region r_i, r_j is the most similar region adjacent to region r_i, and μ_{r_i} and μ_{r_j} are the mean of subregional space-mean density within two regions, respectively.

At each partitioning time point, multiple subregion movements are allowed to minimize the CV and NS. This process can be roughly defined as a Markov Decision Process (MDP), as the next decision depends only on the current state (road network partition) and action (subregion movement), rather than on the historical states. This MDP can be represented by a tuple $(\mathscr{S}, \mathscr{A}, \mathscr{P}, \Lambda)$:

- $\mathscr{S}$ represents the complete configuration of the road network system at any decision time point, which can be expressed as $\mathscr{s} = (\mathscr{R}, \mathscr{K}, \mathscr{B}, \mathscr{H}) \in \mathscr{S}$, where $\mathscr{R} = \{r_1, r_2, \cdots, r_{|\mathscr{R}|}\}$ denotes the current road network partition configuration, with each r_i representing a cluster formed by a set of subregions; $\mathscr{K}$ records the mean densities of all current subregions; $\mathscr{B}$ contains the indices of the boundary subregions under the current partition; and $\mathscr{H}$ tracks the recent movement history of all subregions.

- Given state $\mathscr{s}$, the action space $\mathscr{A}(\mathscr{s})$ is defined as all feasible boundary subregion movement operations:

$$A(\mathscr{s}) = \{(\mathscr{b}, r) | \mathscr{b} \in \mathscr{B}, r \in N(\mathscr{b}, \mathscr{R}), Valid(\mathscr{b}, r, \mathscr{R})\}, \tag{30}$$

where $\mathscr{b}$ represents the boundary subregion to be moved, r denotes the target region, and $N(\mathscr{b}, \mathscr{R})$ is the set of adjacent regions at the current location of subregion $\mathscr{b}$. $Valid(\mathscr{b}, r, \mathscr{R})$ is a binary function that returns true if and only if, after moving $\mathscr{b}$ to region r, all regions remain spatially connected and each cluster contains at least one subregion.

- $\mathscr{P}$ represents the state transition function. Since our problem is a deterministic game, the probability of transitioning to a new state $\tilde{\mathscr{s}}$ is either 0 or 1.
- $\Lambda(\mathscr{s}, \mathscr{a}, \tilde{\mathscr{s}})$ is the function that computes the immediate reward of taking action $\mathscr{a}$ and reaching state $\tilde{\mathscr{s}}$, which can be formally defined as follows:

$$\Lambda(\mathscr{s}, \mathscr{a}, \tilde{\mathscr{s}}) = \underbrace{\gamma \cdot \Delta CV(\mathscr{s}, \tilde{\mathscr{s}})}_{\text{Homogeneity}} + \underbrace{(1-\gamma) \cdot \Delta NS(\mathscr{s}, \tilde{\mathscr{s}})}_{\text{Separation}} - I_{V\text{Constraint penalty}}(\tilde{\mathscr{s}}), \tag{31}$$

where $\Delta CV(\mathscr{s}, \tilde{\mathscr{s}})$ represents the change in the CV, which includes both the measure of the decrease in the number of regions with a CV value exceeding a certain threshold δ, as well as the measure of the decrease in the average CV value. $\Delta NS(\mathscr{s}, \tilde{\mathscr{s}})$ denotes the extent of the decrease in the average NS, and $I_{V\text{Constraint penalty}}(\tilde{\mathscr{s}})$ represents the penalty for violating the region connectivity and size constraints. γ is a tuning parameter that governs the preference for different objectives at each phase of the process. Our model adopts a multi-phase reward function construction. During the CV optimization phase, γ is set to 1. When further optimization of CV is no longer feasible, the model transitions to the NS optimization phase. In this phase, the optimization process must ensure that no deterioration in CV occurs.

This study proposes an MCTS algorithm to solve the above MDP problem. MCTS was initially introduced for a computer Go program (Coulom, 2006; Kocsis and Szepesvári, 2006) and quickly gained popularity in planning or game problems involving sequential decision-making (interested readers may refer to detailed reviews in Browne et al., 2012; Świechowski et al., 2023). The algorithm starts from a root node and constructs a partial, asymmetric game tree, where the child (or leaf) nodes are generated by executing an action from the parent node (Browne et al., 2012). It uses random simulations to estimate the value of actions, and these values guide the generation of the optimal strategy (Browne et al., 2012). Each iteration of the vanilla MCTS algorithm consists of four phases: selection, expansion, simulation, and backpropagation (Świechowski et al., 2023). Specifically, in the selection phase, the algorithm starts from the root node (initial state) and recursively selects child nodes according to a defined strategy until it reaches a node with unvisited (expanded) child nodes or the problem reaches a terminal state. In our problem, we employ an improved version of the classic upper confidence bound (UCB1) method to evaluate child nodes and introduce a random search mechanism to avoid local optima (with extremely low probability of randomly selecting child nodes). The UCB1 method selects the action to take by balancing the exploration and exploitation:

$$\mathscr{a}^{*} = \underset{\mathscr{a} \in \mathscr{A}(\mathscr{s})}{\arg\max} \left\{ \mathscr{Q}(\mathscr{s}, \mathscr{a}) + \omega \sqrt{\frac{2 \cdot \ln[\mathscr{N}(\mathscr{s})]}{\mathscr{N}(\mathscr{s}, \mathscr{a})}} \right\}, \tag{32}$$

where $\mathscr{Q}(\mathscr{s}, \mathscr{a})$ is the average rewards of performing action $\mathscr{a}$ in state $\mathscr{s}$ in the simulation, $\mathscr{N}(\mathscr{s})$ represents the number of simulation times, $\mathscr{N}(\mathscr{s}, \mathscr{a})$ denotes the number of times action $\mathscr{a}$ was performed, and ω is a trade-off parameter that is dynamically adjusted. If the terminal state is not reached, the expansion phase executes a feasible action to expand the tree, generating a new child node. Our model prioritizes selecting the highest-rated untried actions, while also incorporating a lower probability of random selection to enhance diversity. In the simulation phase, starting from the new node, a random simulation strategy is employed to generate a series of results, known as a rollout, which continues until the maximum number of simulation iterations or an early termination condition is met. The backpropagation phase then propagates the payoffs along the search path from the leaf node to the root node, updating the statistics of each node (average reward and visit count). The final optimization result corresponds to the optimal action computed using Eq. (32) in the last iteration. During the early stages of MCTS algorithm iteration, greater emphasis is placed on reducing the CV metric, aiming to decrease the number of problematic clusters in the road network where the CV exceeds the threshold and to optimize the overall average CV value. In the later stages, further optimization is focused on improving the NS value of the partition, building upon the progress made in the earlier phase. To prevent the repeated movement of the same subregion, the algorithm also maintains a Tabu list to record recently moved subregions, applying a penalty to those subregions that have been moved recently. The detailed pseudocode of the MCTS algorithm can be found in Appendix A.3.

Each time the MCTS algorithm identifies the optimal subregion movement at the current iteration, the new state resulting from that movement is treated as the initial state to continue searching for the optimal action until the maximum iteration limit is reached. As a result, the final optimal road network partition at the current time point may involve multiple subregion movements. Nevertheless, the model will always return the best partition solution from all historical iterations.

3. Two-level static partitioning on an urban network

To evaluate the effectiveness of the two-level regionalization approach, this section presents static road network partitioning

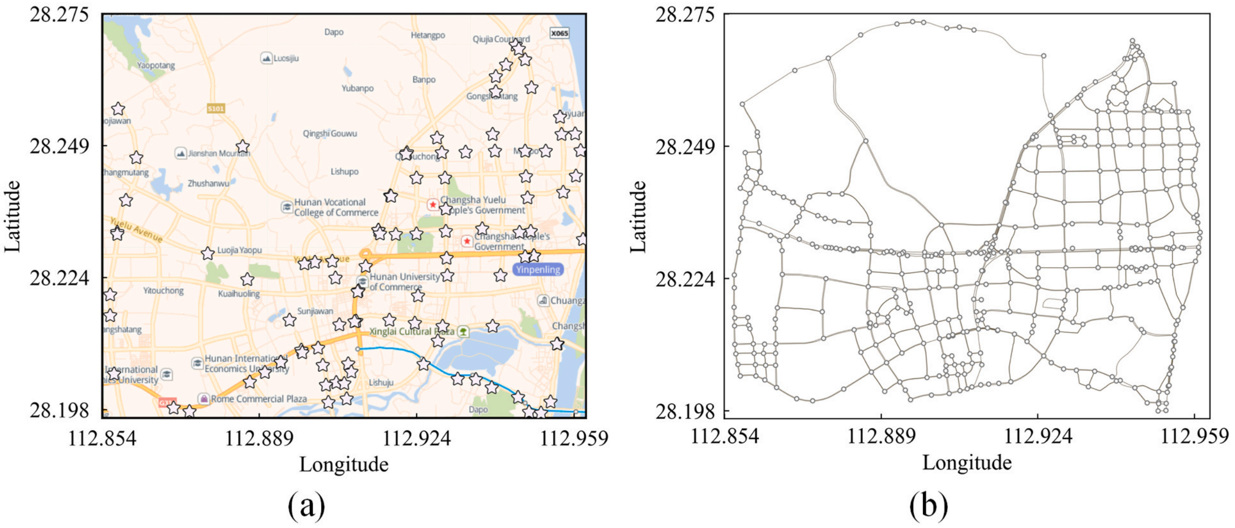


Fig. 3. (a) Spatial distribution of detectors (marked as pentagrams); (b) link-level representation.

Table 3
Partitioning performance under varying homogeneity weights (mean/standard deviation).

Homogeneity weights	Subregion-level		Region-level	
	TV_n	**Boundary edge ratio**	***TV_n***	**Boundary edge ratio**
0.1	0.859/0.019	0.118/0.003	0.956/0.017	0.045/0.005
0.3	0.851/0.013	0.118/0.002	0.942/0.017	0.041/0.005
0.5	0.804/0.022	0.131/0.010	0.927/0.027	0.056/0.008

experiments on a real urban road network. While previous theoretical studies typically assumed a regular "subregion-region" system (e.g., hexagonal grids), this section aims to analyze the feasibility of constructing such a system on irregular road networks.

The road network used is part of the Yuelu District of Changsha City, Hunan Province, China. The original network comprises 3,227 intersections and 7,118 links, with a total length of 93.8 km. The raw data were collected by 89 automatic license plate recognition systems over a three-day period from October 19 to October 21, 2022. Incomplete vehicle trajectories were constructed by matching license plate records, which captures vehicle passage. To reconstruct the paths between non-adjacent trajectory points, we employed the path reconstruction method described in Lopez et al. (2017a,b). A simplified road network that preserves the primary topological structure was extracted by excluding unattractive links, i.e., those rarely traversed by vehicles, identified by a set of complete vehicle trajectories. This refined network consists of 419 intersections and 994 links, with a total road length of 39.3 km. Fig. 3(a) illustrates the spatial distribution of detectors within the research area, while Fig. 3(b) presents the extracted road network. The link mean speeds during the entire morning peak period are used as the congestion measure, which serve as input for static road network partitioning. The calculation of link speed profiles follows the method outlined by Lopez et al. (2017a,b). Specifically, for links equipped with ALPR systems at both upstream and downstream ends, anomalous vehicle travel time records are filtered out before computing the mean speed. For links on the reconstructed path, the speed records are approximated by the average speed across the entire path. This approach enables the calculation of mean speed on most links in the road network during the morning peak period. Then, the heuristic approach presented by Gu and Saberi (2019) was employed to estimate the mean speed on the remaining links with missing data.

3.1. Sensitivity analysis

Given that the proposed regionalization models are inherently multi-objective optimization problems, this section conducts a sensitivity analysis of the trade-off weights within the model. These weights determine the homogeneity and compactness of the subregions and regions. The weight of the compactness metric (boundary edge ratio), i.e., λ_2 and λ_4, is fixed at 1 for both regionalization models, while the impact of varying the weight of the homogeneity metric (λ_1 and λ_3) on partitioning performance is analyzed. Furthermore, we aim to validate the ability of this two-step regionalization approach to achieve the desired road network partitioning outcomes, as it represents the first method to account for different partitioning granularities.

For simplicity, the weight of the homogeneity metric is set to be consistent in both models, taking values of 0.1, 0.3, and 0.5. In addition, the minimum number of links for subregions is set to 50, and the minimum number of subregions per region is set to three. The number of regions is set to four. The remaining model parameters are detailed in Appendix A.4. All algorithms were implemented in Python and executed on a computer equipped with a 2.6 GHz Intel Core i7 processor and 16 GB of RAM. The MILP model was solved

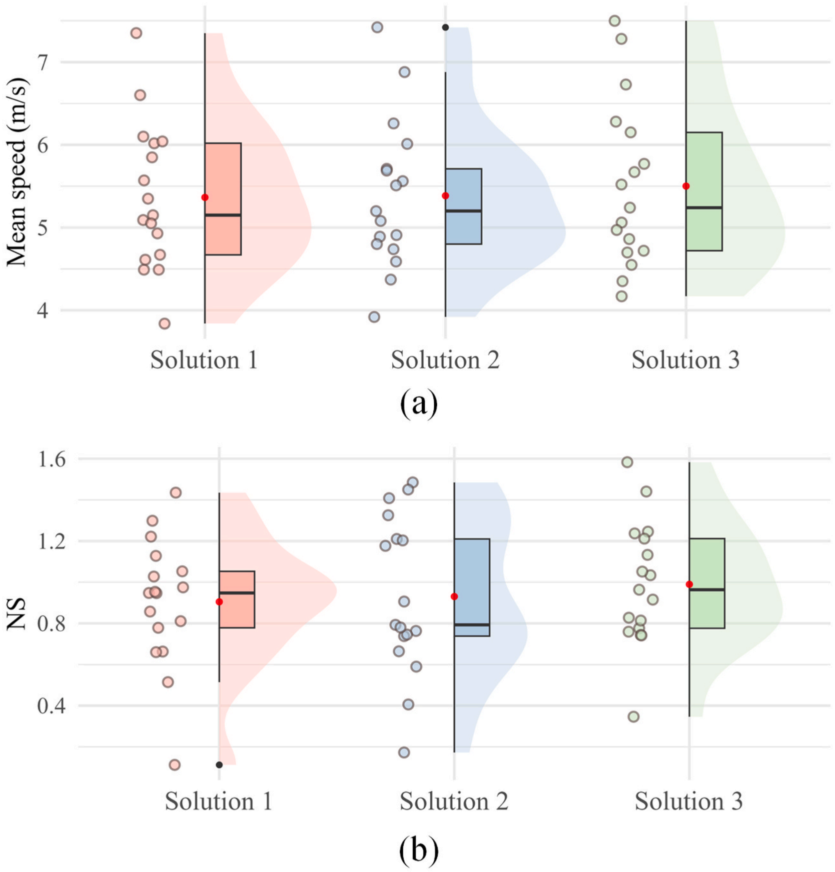


Fig. 4. (a) Mean speed distribution and (b) *NS* distribution of subregions.

Table 4
Regional indicators under different partitioning solutions (mean/standard deviation of link speeds).

	Region 1	Region 2	Region 3	Region 4	TV_n
Solution 1	5.160/2.170	5.610/2.490	5.070/1.790	6.120/2.530	0.971
Solution 2	5.290/1.670	5.630/2.450	4.610/1.760	6.280/2.590	0.936
Solution 3	5.220/2.340	5.620/2.000	4.430/1.570	6.590/3.120	0.919

using the open-source COIN-OR Branch-and-Cut solver (Lougee-Heimer, 2003), which is integrated in a Python library named PuLP (Mitchell et al., 2011).

The maximum number of subregions, subject to internal threshold constraints, was determined to be 17 through 1000 iterations of region growth and enclave assignment. Subsequently, for each homogeneity weight, ten trials were conducted based on the initial partitioning solution, yielding ten partitioning solutions at both the subregion and region levels. Table 3 presents the performance of the subregion and region level partitioning under different weightings. It can be observed that, in all cases, the standard deviations of TV_n and boundary edge ratio remain relatively low, reflecting the stability of the proposed algorithm. The changes in the two metrics at the average level are generally reasonable, with TV_n decreasing at both levels as more weight is given to homogeneity, while the boundary edge ratio gradually increases.

Given that the primary objective in partitioning subregions is to identify pockets with significant variations in congestion levels, we further analyzed the degree of differentiation between subregions under different weights. One solution out of every 10 trials was selected for detailed analysis, with solution 1 corresponding to a homogeneity weight of 0.1 and solution 3 corresponding to a homogeneity weight of 0.5. Fig. 4 illustrates the distribution of the mean speeds and *NS* values of the subregions for the three solutions. It is evident that the mean speeds of the subregions in all three solutions exhibit a relatively dispersed distribution, suggesting that the ANLS algorithm can effectively identify local congestion clusters under different weights. In addition, the differences in the distributions across the three solutions are minimal, confirming the hypothesis that subregions can be both homogeneous and compact with a sufficiently small partitioning granularity. However, the distribution of *NS* values shows that a significant proportion of subregions have *NS* values greater than 1. This is reasonable, since at finer granularities the congestion levels between adjacent subregions often

Table 5
Method comparison results.

	TV_n	NS	Boundary edge ratio
GCL-FINCH	0.925	0.946	0.088
SNMF	0.919	0.981	0.195
Proposed model-ANLS	0.933	0.950	0.054
Proposed model-MILP	0.919	0.979	0.062

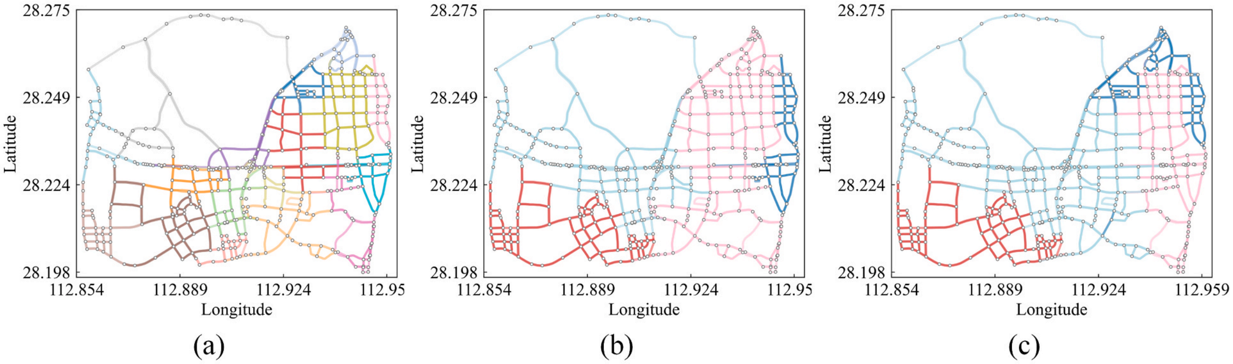


Fig. 5. (a) Subregion-level partitioning result; (b) region-level partitioning result (ALNS); (c) region-level partitioning result (MILP).

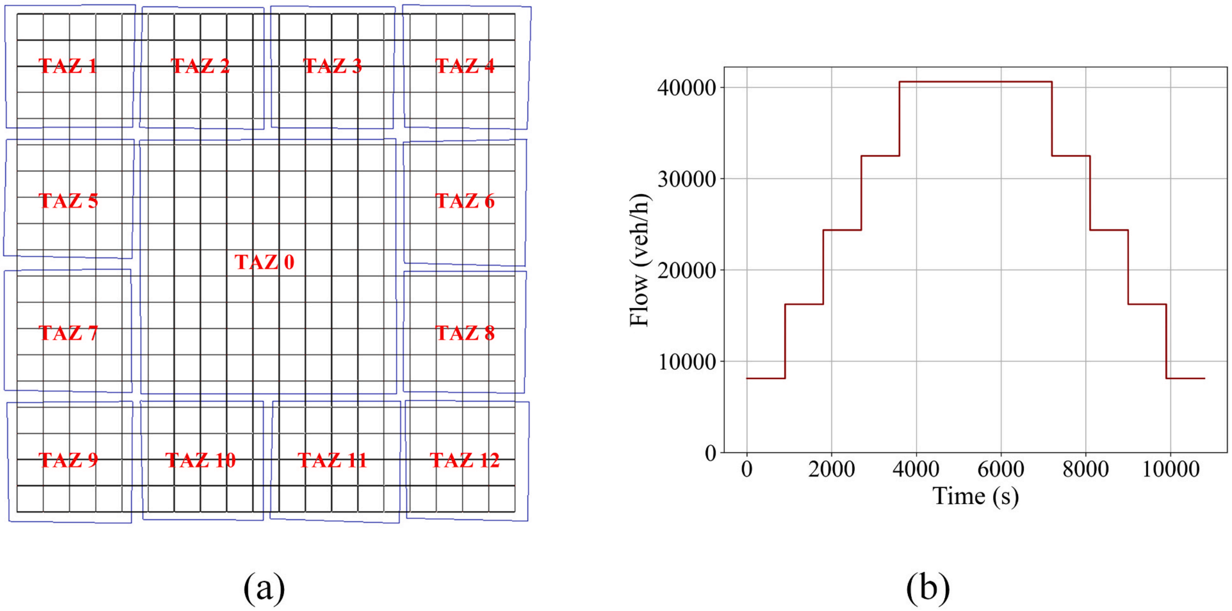


Fig. 6. (a) Traffic analysis zones; (b) dynamic demand scheme.

change gradually. Furthermore, the ability of subregions to absorb similar links from adjacent subregions is limited due to the size constraints and the model's mechanism for maximizing the number of subregions. Table 4 presents the performance of the three solutions at the region level. After aggregating adjacent subregions, the congestion level differences between regions become smaller. As the homogeneity weight increases, the maximum difference in mean speeds between regions also increases, from 1.05 to 2.16, while TV_n decreases from 0.971 to 0.919. This suggests that the linear homogeneity objective in the region-level model is effective and yields results consistent with TV_n.

3.2. Comparison results

Despite the fact that the improvement of the static road network partitioning performance is not the main focus of this study, this

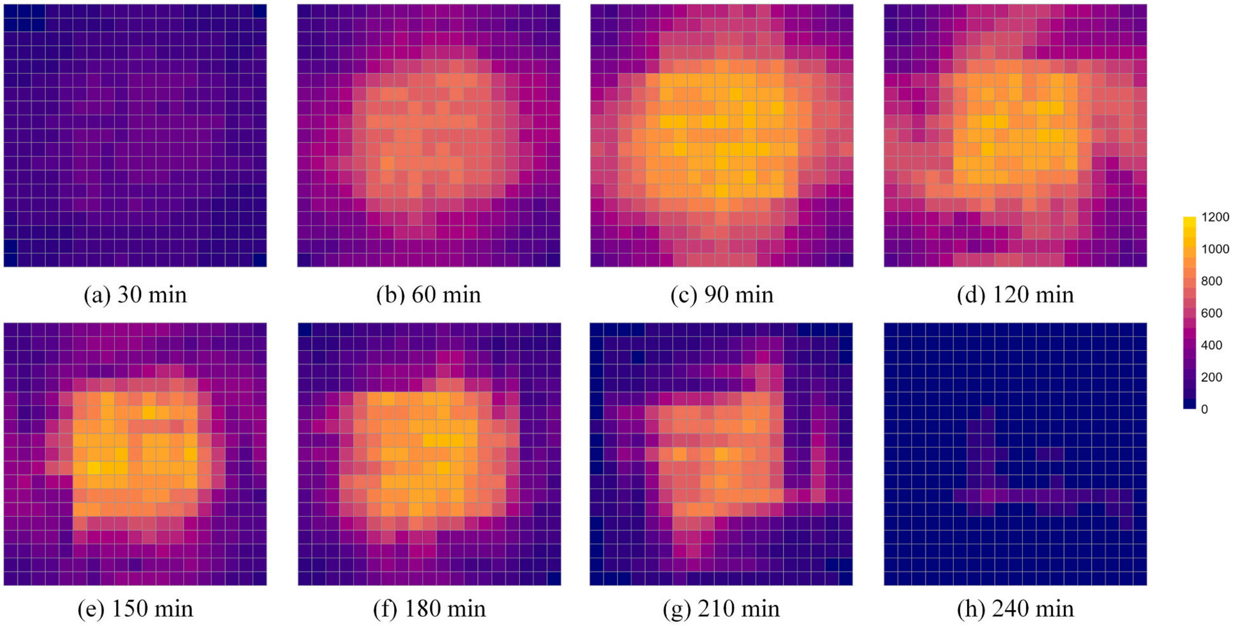


Fig. 7. Heatmap of traffic flow distribution at different timestamps (veh/h).

section aims to briefly validate the feasibility of the proposed two-step regionalization approach in achieving a performance comparable to that of alternative methods. The benchmark methods used for comparison include the combination of the SNMF algorithm and the "snake" algorithm from Saeedmanesh and Geroliminis (2016) (referred to as SNMF), as well as the framework proposed by Hu et al. (2025b) that integrates graph contrastive learning with hierarchical clustering (first integer neighbor clustering hierarchy, FINCH), referred to as GCL-FINCH. In addition, we considered a variant where the ALNS algorithm from the first-level regionalization model is directly applied to solve the second-level regionalization problem (with the homogeneity metric remaining as TV_n).

Table 5 presents the specific comparison results for a region count of 4. It is important to note that the results for SNMF and GCL-FINCH in Table 5 place greater emphasis on compactness, and the results for GCL-FINCH reflect the average across multiple trials. As shown, the proposed model (Proposed model-MILP) achieves the lowest TV_n while maintaining the second lowest boundary edge ratio. Although the model that applies ANLS in both levels (Proposed model-ANLS) performs best in terms of compactness, it exhibits poorer homogeneity. Overall, the proposed model provides superior static road network partitioning solutions, which can serve as the foundation for initiating the dynamic road network partitioning process. Fig. 5 illustrates the subregion partitioning results when the homogeneity weight is set to 0.5, along with the regional partitioning results obtained using ALNS and MILP. It is evident that both subregions and regions can maintain compactness while preserving a reasonable degree of homogeneity within the subregions/regions.

4. Dynamic mechanism analysis on a simulated grid network

This section examines the influence of subregion size on maintaining temporal homogeneity and validates the effectiveness of the proposed dynamic updating mechanism using a simulation grid network with 20 × 20 nodes developed in SUMO. In this network, each pair of adjacent intersections is connected by two one-way streets. All intersections are signal-controlled, utilizing SUMO's default gap-based actuated traffic control. Fig. 6(a) shows the traffic analysis zone (TAZ) partitioning on the grid network, which consists of 13 TAZs, with TAZ 0 located at the center of the network and the remaining TAZs representing peripheral areas. To simulate the morning peak period in a monocentric city, an asymmetric origin–destination (OD) demand matrix was designed, with the majority of traffic flowing from the peripheral areas toward the central area (TAZ 0). The basic demand profile can be found in Appendix A.5. The loading and unloading phases of the network are constructed by adjusting the scaling factor. The total simulation duration is four hours, with the final hour serving as the network's clearance time, during which no additional vehicles are generated. The overall dynamic demand flow profile is illustrated in Fig. 6(b). Loop detectors are installed on each road (positioned at the midpoint to prevent overestimation of traffic density) and record vehicle count, occupancy, and average speed at regular time intervals. A total of 81,200 vehicles are generated into the network throughout the simulation period.

Fig. 7 presents the traffic flow distribution across the grid network at different timestamps. It is evident that the network loading process continues until the 120th minute, and significant congestion begins to develop in the central region of the network starting from the 60th minute. As the network unloads, the congestion clusters in the center gradually diminish, and finally disappear by the 240th minute. The average link density over the entire simulation period is used as the historical data input. A two-level regionalization approach is then applied to generate the static road network partitioning results for the network, as shown in Fig. 8. The

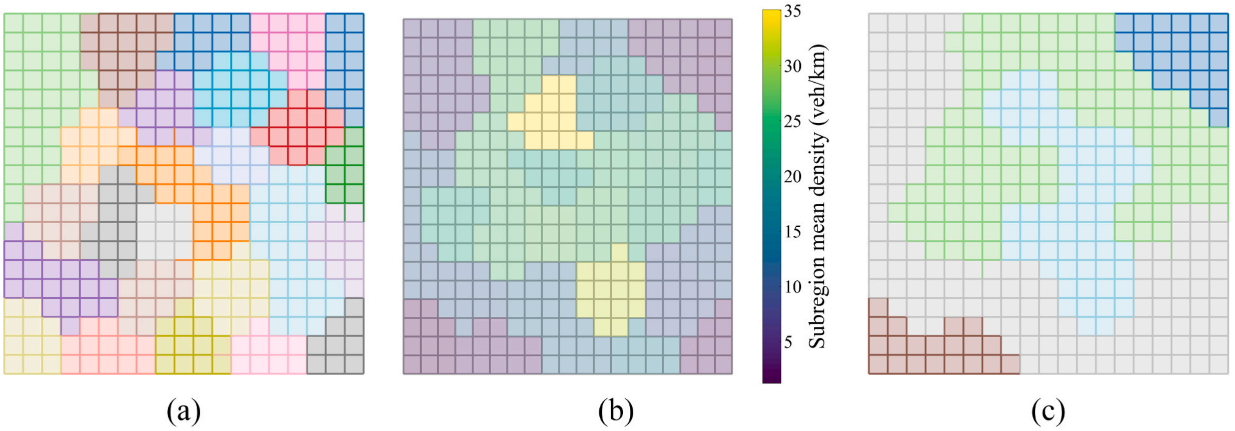


Fig. 8. (a) Subregion-level partitioning result; (b) heatmap of subregional mean density; (c) region-level partitioning result.

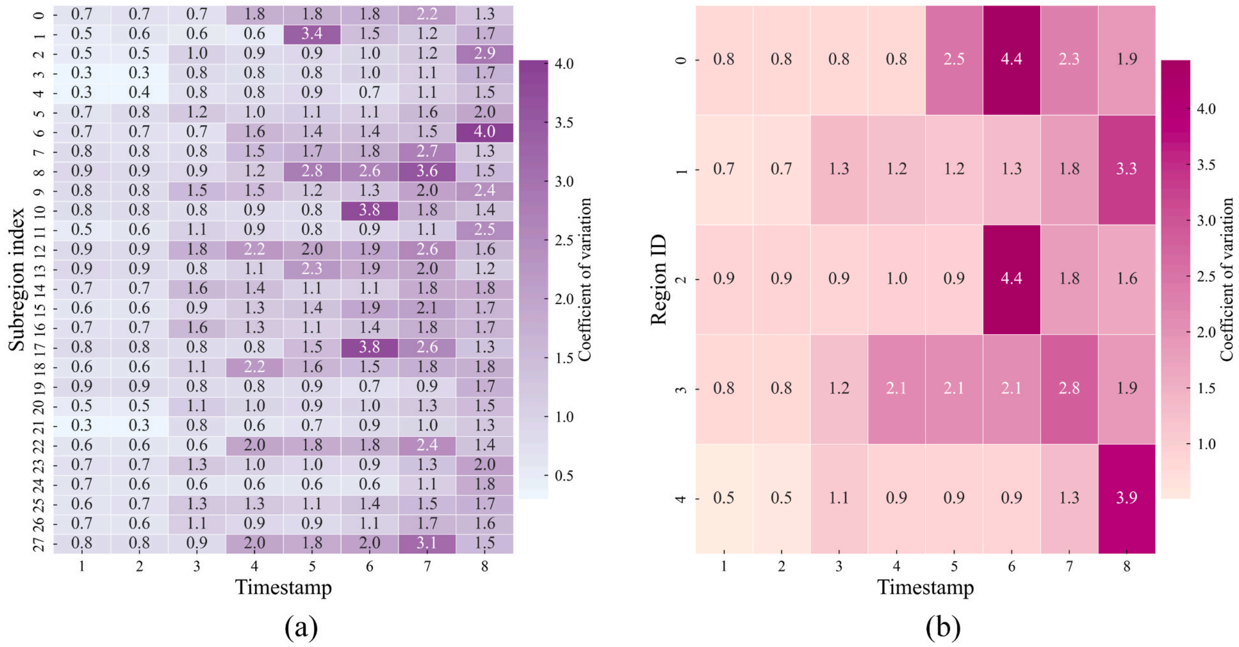


Fig. 9. Coefficient of variation of (a) subregions and (b) regions within different time periods.

parameters used in the model, with the exception of adjusting the number of regions to five and setting the threshold for the number of subregions within each region to two, remain consistent with those used in the previous experiments.

As shown in Fig. 8(a), 28 compact and approximately homogeneous subregions are identified within the road network. The distribution of subregional mean density is shown in Fig. 8(b). It is evident that two most congested subregions are located near the center of the network. Almost all subregions show considerable differences in congestion levels compared to some of their adjacent subregions. Fig. 8(c) illustrates the region-level partitioning results. It can be seen that the two most congested subregions, along with several slightly congested adjacent subregions, are grouped into a single region (represented in blue). The remaining subregions with comparatively lower congestion surrounding the central area are assigned to another region (represented in green). As shown in Fig. 8 (b), the least congested subregions are located at the four corners of the network. The region-level regionalization model groups the two subregions in the lower left corner and the two in the upper right corner into two separate regions, while the remaining less congested subregions are grouped into a single region. This is due to the fact that the subregions in the other two corners are isolated, preventing them from forming a region that meets the subregion threshold. This result demonstrates the applicability of the proposed model to road networks with nested congestion distributions, a characteristic that many graph-based spectral clustering methods struggle to achieve.

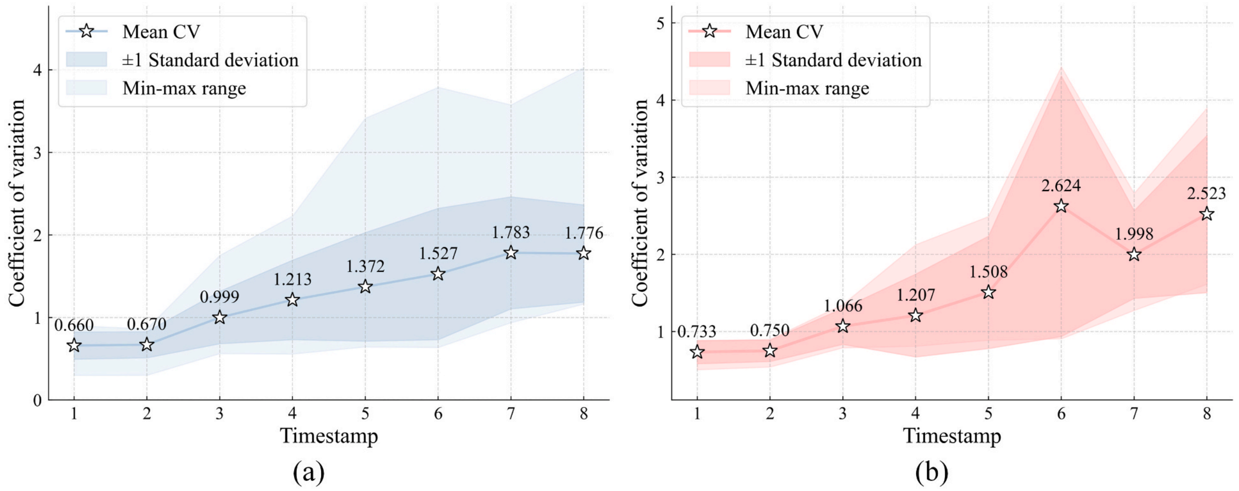


Fig. 10. Temporal evolution of the coefficient of variation: (a) subregion-level partitioning result; (b) region-level partitioning result.

4.1. Temporal variation analysis

Once the "subregion-region" system is established, a critical question arises: to what extent does a system based on historical average data maintain temporal stability? In other words, can the homogeneity within these subregions and regions endure over time? Since the proposed method defines the subregions as fixed, this section investigates the temporal variation in the level of homogeneity within these subregions. For the region-level partitioning result, a parallel analysis is performed in which the region is treated as the result of a higher threshold for subregional partitioning (i.e., fewer subregions). The purpose of this approach is to examine the influence of partitioning granularity on the temporal stability of subregions.

Using a 30-minute interval for data collection by the detector, link mean density data were obtained for eight time intervals. The coefficient of variation (*CV*) of the density within subregions is used as an indicator of homogeneity, with lower values reflecting greater homogeneity. Fig. 9(a) shows a heat map of the *CV* for subregions over different time intervals. It is observed that during the initial loading phase of the road network (the first two time intervals), the *CV* within subregions remains relatively low (ranging from a minimum of 0.3 to a maximum of 0.9). As additional vehicles enter the network, the road network becomes increasingly heterogeneous, leading to a continuous increase in *CV* for most subregions, with some reaching values as high as 4. However, it is noteworthy that a significant proportion of subregions do not experience a *CV* exceeding 2 over time. In contrast, almost all regions exhibit a *CV* above 3 in the later stages of the simulation, although their initial *CV* is below 1, as shown in Fig. 9(b).

We further analyzed the temporal variation in the homogeneity of subregions and regions over different time intervals. As shown in Fig. 10, in addition to the gradual increase in the mean *CV* for both subregions and regions over time, the standard deviation of the *CV* and the range between the minimum and maximum values also increase. This suggests that some subregions and regions exhibit deviations in temporal stability, while others demonstrate relatively better temporal stability. In particular, this intra-group variability becomes more pronounced during the road network unloading phase. For almost all time intervals, the average homogeneity at the subregional level is better than at the regional level, which is reasonable since each region is composed of a smaller number of less homogeneous subregions.

Overall, the subregions created based on historical data demonstrate reliable homogeneity in the initial phase of road network loading. However, the homogeneity of these subregions fluctuates over time. As the road network enters a phase of extreme congestion, the homogeneity within a small number of subregions deteriorates significantly. Nevertheless, this finer-grained partition proves advantageous in terms of improving the temporal stability of individual units (i.e., subregions or regions). This trade-off, however, is a necessary compromise when aligning the empirical investigation with theoretical perimeter control research.

4.2. Dynamic update results

Based on dynamic traffic flow data derived from the simulation outputs, this section validates the effectiveness of the proposed dynamic road network partitioning method. A 30-minute interval is maintained between successive partition updates. In the MCTS algorithm, the maximum number of iterations for the outer loop is set to 20, the simulation phase is configured to run for 100 iterations, the maximum simulation depth is set to 8, the maximum number of consecutive moves per iteration is set to 5, and the initial weighting parameter is set to 1.1. The coefficient of variation threshold for defining a relatively homogeneous region is set to 0.3. Using the new data collected every 30 min, the proposed MCTS algorithm continuously updates the current road network partition to adapt to the evolving congestion distributions. It is important to note that, at this stage, the focus is on the space-mean density within each subregion, rather than the link-level density. Fig. 11(a-h) present the partition solutions obtained by the dynamic road network partitioning algorithm at different time intervals. It can be seen that, based on the data from the first two time intervals, the updated

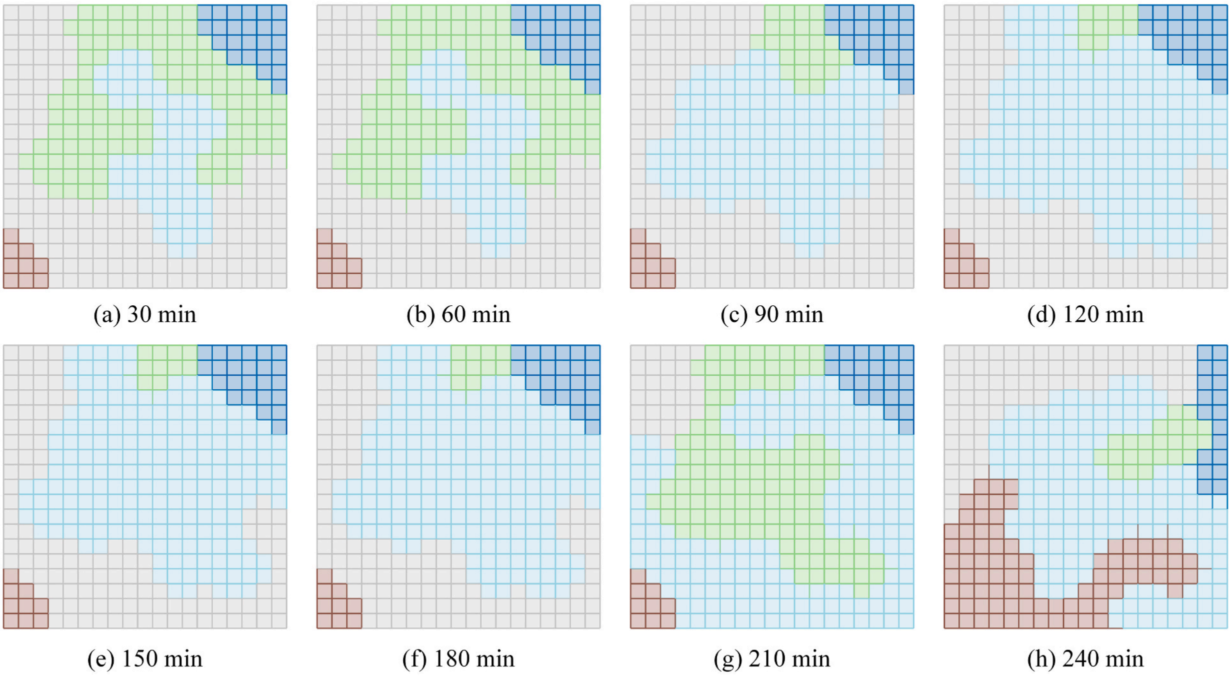


Fig. 11. Dynamic road network partitioning results.

road network partition solution is almost identical to the initial (static) partition. This similarity is due to the fact that during the initial loading phase of the road network, the *CV* within all regions remains at a relatively low level. As the congested area within the network expands and contracts, partition solutions begin to emerge that differ significantly from the initial configuration. This shift occurs because the dynamic road network partitioning algorithm adapts to changing traffic conditions by adjusting the boundaries of subregions in multiple steps. Fig. 11(a-f) clearly show the gradual expansion of the congested region.

We further compared the time-respected homogeneity performance under both dynamic and static road network partitioning schemes. Fig. 12 shows the spatiotemporal heatmaps of *CV* and *NS*. A value of 0 indicates that the corresponding region at a given time interval consists of only one single subregion. It can be observed that, under the dynamic road network partitioning, a region composed of a single subregion appears during the first three time intervals. In addition, from the 4th to the 6th interval, an extra region of the same type is generated. In the dynamic scheme, the *CV* of all regions remains below the threshold of 0.3 during the first three intervals. In contrast, in the static partitioning scheme, one or more regions consistently exceed this threshold. This difference is particularly pronounced in the last time interval, when all but one region in the static scheme exhibits *CV* values above 0.4. Conversely, only one region in the dynamic scheme has a significantly high *CV*. Across all time intervals, the number of regions satisfying the *CV* threshold is consistently higher in the dynamic scheme than in the static scheme. In terms of *NS*, the dynamic partitioning scheme also demonstrates improved performance, reducing the *NS* in most cases. For example, in the last time interval, four regions under the dynamic scheme have *NS* values not exceeding 0.2, whereas the static scheme includes three regions with *NS* values exceeding 1.

Fig. 13 presents the variation trends of the average *CV* and *NS* values of the road network under both schemes (excluding regions composed of a single subregion in the calculations). It is evident that, throughout all time intervals, the average *CV* of the road network under the dynamic road network partitioning scheme remains consistently lower than that under the static road network partitioning scheme. In addition, the average *NS* values for both schemes are quite similar during the first five time intervals. However, the average *NS* under the dynamic scheme remains consistently within a reasonable range, while the average *NS* under the static scheme reaches values close to 0.9.

In summary, the proposed dynamic road network partitioning algorithm proves to be highly effective in reducing both *CV* and *NS* of regions. This ensures both dynamic homogeneity within regions and clear differentiation between regions over time, thereby facilitating the identification of boundaries that require control in perimeter-based traffic management strategies. Although this study primarily considers *CV* and *NS* as the optimization objectives for dynamic road network partitioning, other objectives can easily be integrated into the MCTS algorithm to achieve the desired results.

5. Dynamic partitioning validation on a real road network

This section further validates the comprehensive framework using a medium-scale real urban road network. As depicted in Fig. 14 (a), the researched network corresponds to the simulated urban road network of Bilbao, Spain, as provided by Aimsun (Aimsun, 2024). This network consists of 571 intersections and 1,263 road segments, spanning a total road length of 156.22 km. A calibrated dynamic

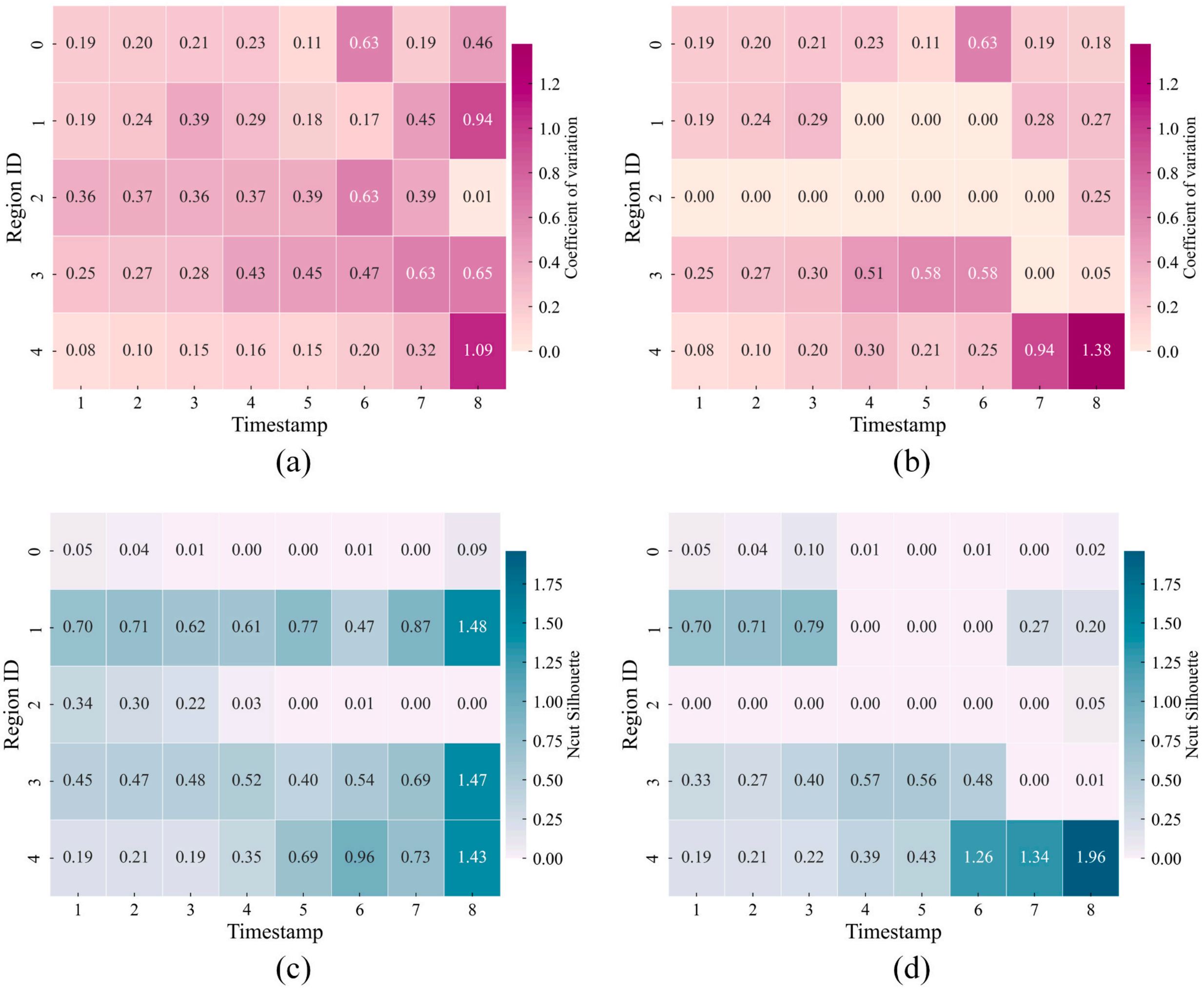


Fig. 12. Spatiotemporal heatmaps of *CV* and *NS* for static solution (a, c) and dynamic solution (b, d).

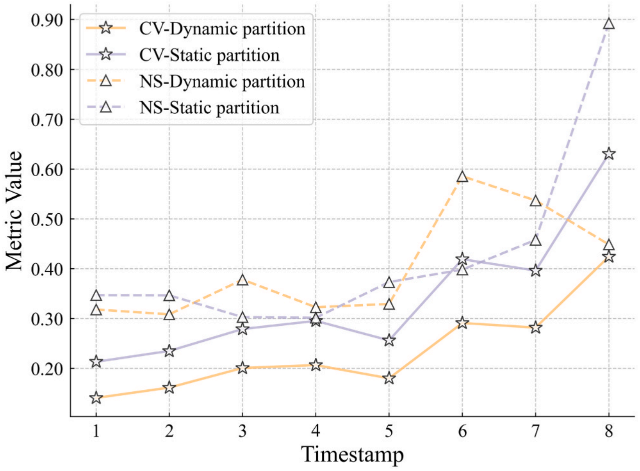


Fig. 13. Comparison of overall performance between static and dynamic partitions.

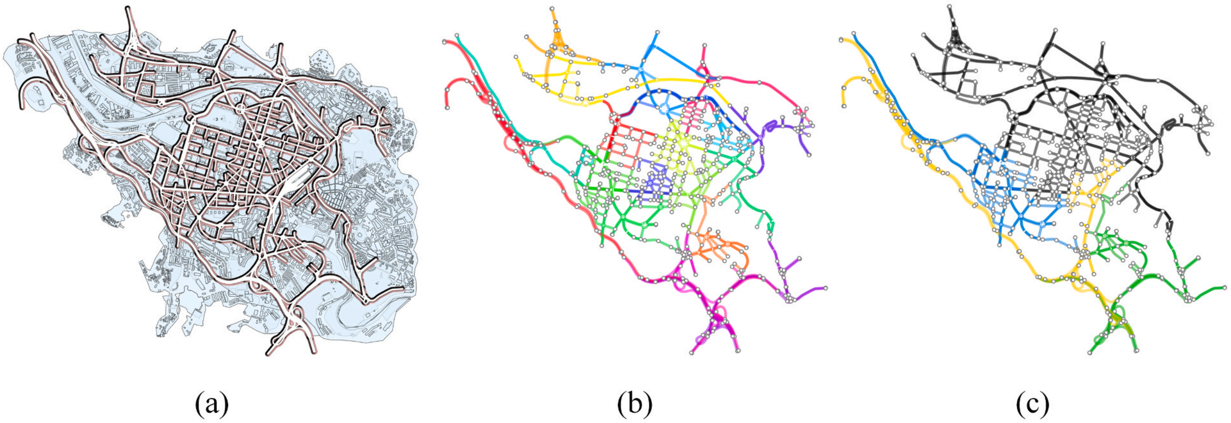


Fig. 14. (a) Bilbao road network; (b) subregion-level partitioning result (20 subregions); (c) region-level partitioning result.

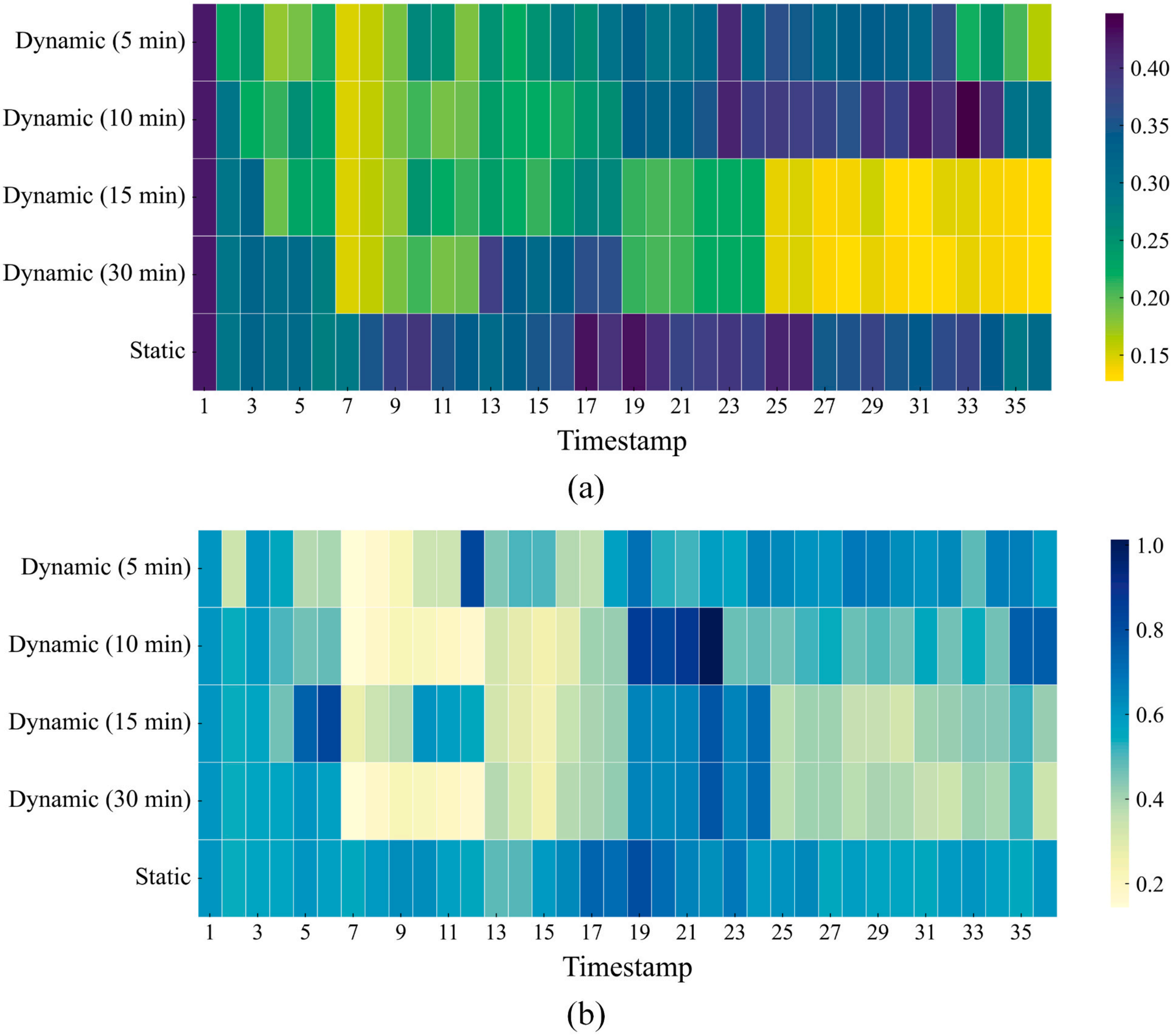
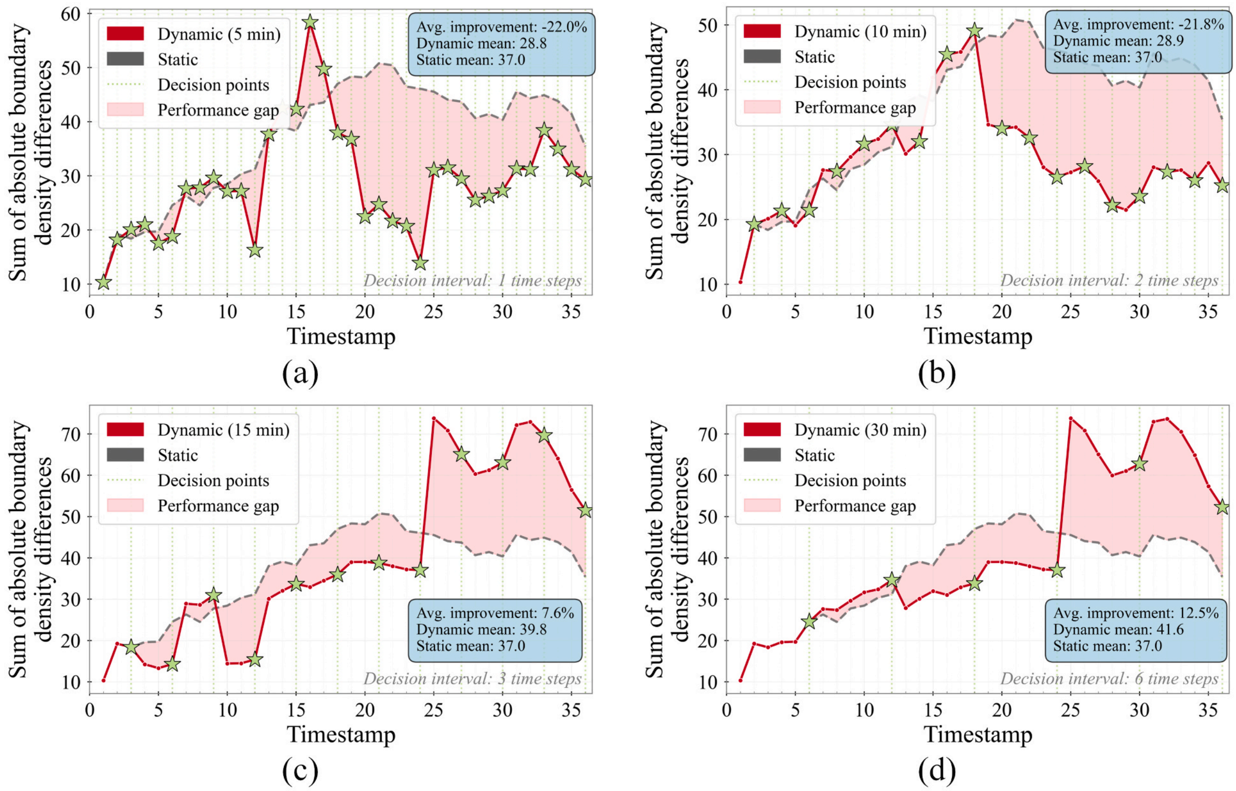


Fig. 15. Heatmaps of (a) average *CV* and (b) average *NS* over time.



(a) (b) (c) (d)

Fig. 16. Performance comparison of the sum of absolute boundary density differences between (a-d) dynamic partitioning with decision time intervals of 5, 10, 15, and 30 min and static partitioning.

origin–destination matrix is provided for the morning peak hour and scaled by factors of $[0.8, 1, 0.9]$ to extend it over three hours, thereby simulating the entire morning peak. A total of 20,984, 26,230, and 23,607 vehicles entered the network during each hour, respectively. The simulation experiment provides average link data over the entire simulation period, as well as dynamic link data collected every five minutes. Similarly, the average link density during the simulation is used as historical data to execute two-level static network partitioning. The weights of the homogeneity and compactness metrics (i.e., λ_1, λ_2, λ_3, and λ_4) in the regionalization model are both set to 1, and the other parameters and computational equipment are consistent with those specified in Section 3.1. The number of regions is set to four, and the MILP model is solved using the COIN-OR Branch-and-Cut solver. Fig. 14(b) shows that the max-p region algorithm identifies 20 compact and nearly homogeneous subregions. Fig. 14(c) presents the optimal region-level partition solution. The solution times for the two regionalization models were 34 and 38 s, respectively, with a total computation time of 72 s.

Subsequently, dynamic updates of the network partitioning were conducted with decision time intervals of 5, 10, 15, and 30 min. To closely replicate the application scenario of dynamic network partitioning, it is assumed that, at each decision time point, the MCTS algorithm optimizes the previous network partitioning solution based solely on link data collected from the preceding time step (i.e., five minutes) rather than utilizing all data between two decision time points. This solution is then carried over to the next decision time point. This approach significantly reduces the data collection workload and conserves storage resources in practical applications. The parameters of the MCTS algorithm remain consistent with those outlined in Section 4.2. Fig. 15 illustrates heatmaps of the time-respected average *CV* and *NS* across all regions of the network under different partitioning schemes. It is important to emphasize that the calculation of *CV* and *NS* in this context accounts for the lag associated with dynamic network partitioning. Specifically, all time steps within a decision interval are calculated based on the network partitioning solution updated at the preceding decision time point. Therefore, in dynamic partitioning scenarios with decision time intervals larger than the time step, the average *CV* and *NS* for several earlier time steps align with those observed in static network partitioning scenarios. The average decision time of the MCTS algorithm across different decision intervals is between seven and eight seconds. As shown in Fig. 15(a), for most time steps, the average *CV* under dynamic network partitioning is lower than that under static network partitioning. Notably, the average *CV* of the network typically decreases significantly under the optimization of the MCTS algorithm for the first time step immediately following a decision time point. However, it can be observed that increasing the decision update frequency does not lead to performance improvements. Specifically, dynamic network partitioning scenarios with longer decision intervals generally exhibit better average *CV* performance then those with shorter decision intervals. This suggests that more frequent network partition updates may lead to a deterioration of homogeneity within regions. The underlying reason may be that the propagation and dissipation of traffic congestion

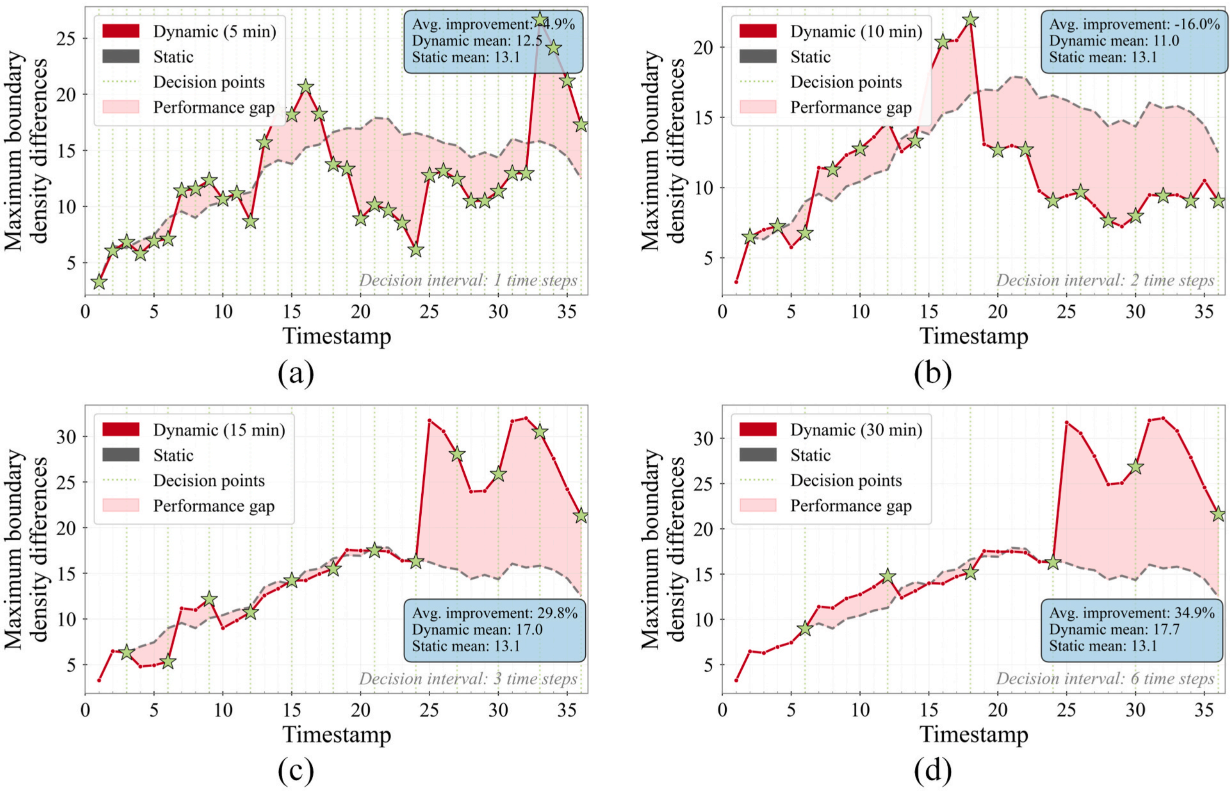


Fig. 17. Performance comparison of the maximum boundary density differences between (a-d) dynamic partitioning with decision time intervals of 5, 10, 15, and 30 min and static partitioning.

require time to be reflected in the link data. Additionally, a five-minute time step may be subject to considerable random fluctuations and measurement noise. Frequent adjustments based on such short-term, volatile data could amplify the effects of noise, thereby diminishing decision quality. Frequent updates could also cause the optimization algorithm to converge to a local optimum. In contrast, decision intervals of 15–30 min are more likely to capture stable traffic patterns, which is beneficial for maintaining homogeneity within regions. Fig. 15(b) demonstrates that the MCTS algorithm with 15- or 30-minute decision intervals consistently yields significantly lower average *NS* values, particularly in the later stages of the simulation (time steps 25–36), further supporting its superiority.

As previously discussed, the dynamic network partitioning algorithm should not only optimize homogeneity within regions but also identify dynamic boundaries that require gating flow control. The greater the difference in the average congestion level between two regions at the boundary, the more advantageous it is to implement perimeter control as it allows for clearer identification of regions requiring protection. This section further analyzes the performance of the sum of absolute boundary density differences (SABDD) and maximum boundary density differences (MBDD) across different network partitioning scenarios. For simplicity, the regional density used to calculate density differences at the boundaries is defined as the arithmetic mean of the network mean densities of the sub-regions within each region. As shown in Fig. 16, for most time steps, SABDD is lower for dynamic network partitioning scenarios with decision intervals of 5 and 10 min than for the static network partitioning scenario, resulting in negative gains of 22.0 % and 21.8 %, respectively. In contrast, longer decision intervals improve SABDD performance. Scenarios with 15- and 30-minute intervals result in average improvements of 7.6 % and 12.5 %, respectively. Fig. 17 further demonstrates that dynamic partitioning with shorter decision intervals leads to lower average MBDD values compared to static partitioning. Conversely, the scenario with a 30-minute decision interval significantly outperforms static partitioning in the majority of time steps, achieving an average MBDD improvement of up to 34.9 %. These findings provide additional evidence that frequent adjustments to regional partitioning may be unsuitable for perimeter control applications.

Overall, experiments conducted on real road networks using high-fidelity simulation data validate the superiority of the proposed dynamic network partitioning framework. Moreover, the results indicate that longer decision intervals are more conducive to maintaining homogeneity within regions and identifying the boundaries of protected regions, thereby facilitating the implementation of perimeter control. Given that the computation time of the MCTS algorithm is considerably shorter than the decision time interval, the scalability of the proposed framework for larger road networks is adequately ensured.

6. Conclusion

This paper proposes a comprehensive dynamic road network partitioning framework to establish a multi-scale network partitioning structure and to adjust this structure in real time to accommodate the evolving distribution of traffic congestion. Two regionalization models are developed to obtain road network partitioning at the subregion and region levels, respectively. A max-p region algorithm combined with an ALNS metaheuristic is used to solve the first regionalization model. Due to the reduced scale of the regionalization problem, the second model can be solved exactly using a solver with modifications to the objective function and connectivity constraints. Subsequently, the problem of updating the road network partitioning for each fixed time interval is formulated as a MDP problem, and a customized MCTS method is developed to solve this problem. The methods of the framework are thoroughly validated using the urban road network of Yuelu District in Changsha, China, a simulated grid road network, and the urban road network of Bilbao. The results demonstrate that the two-level regionalization model yields satisfactory initial partitioning results. Based on the established subregion-region system, the dynamic road network partitioning method with longer decision time interval effectively adapts to heterogeneous congestion distributions by adjusting the boundary subregions.

This paper paves the way for the practical implementation of perimeter-based traffic management strategies integrated with real-time road network partitioning. Future research may focus on improving the efficiency of the algorithm, for example, by using distributed MCTS methods (Świechowski et al., 2023) to accelerate the solution process for dynamic road network partitioning. Furthermore, the integration of the proposed framework with advanced perimeter control methods represents a promising direction for future studies.

CRediT authorship contribution statement

Cheng Hu: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Jinjun Tang:** Writing – review & editing, Supervision. **Junjie Hu:** Software, Methodology. **Yaopeng Wang:** Visualization, Software. **Zhitao Li:** Validation, Methodology. **Jie Zeng:** Validation, Methodology. **Chunyang Han:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded in part by the National Natural Science Foundation of China (No. 52172310), Key R&D Program of Hunan Province (No. 2023GK2014), Science Research Foundation of Hunan Provincial Department of Education (No. 22B0010), Ministry of Education Humanities and Social Science Research (Youth) Project (No. 24YJC630065), Yunnan Fundamental Research Projects (Youth) (No. 202401AU070217), and Furong Scholar Award Program of Hunan Province.

Appendix

A.1. ALNS algorithm

Algorithm 1. Adaptive Large Neighborhood Search

```
Input:
    Initial solution $R_0$
    Temperature schedule parameters $T_0$, $\xi$
    Maximum iterations $I_M$
    Link congestion measurement data $S$
    Trade-off weights $\lambda_1$ and $\lambda_2$
Output:
    Best solution $R^*$
Initialize
$R \leftarrow R_0$, $R^* \leftarrow R_0$
Compute partitioning performance $P(R)$ and $P(R^*)$ by the following equation
$\lambda_1 \cdot \frac{\sum_{r\in R}\left[\sum_{i\in V}\left(\sum_{c\in C} x^{i,r,c}\right)\cdot\left(s^i-\bar{s}^r\right)^2\right]}{\sum_{i\in V}\left(s^i-\bar{s}^V\right)^2} + \lambda_2 \cdot \frac{\sum_{(i,j)\in E}\left(1-\sum_{r\in R} z^{i,j,r}\right)}{|E|}$
Update regional root
Precompute BFS levels for all nodes
Initialize $\alpha$-UCB selector with:
    Destroy operator set $\Phi^-$
    Repair operator set $\Phi^+$
```

(continued on next page)

(*continued*)

Algorithm 1. Adaptive Large Neighborhood Search

```
    Optional coupling matrix $\Psi$
    Score rewards $[3, 2, 1, 0]$
$T \leftarrow T_0$
for $i = 1$ to $I_M$ do
    Select operator pair $(\Phi_d^-, \Phi_r^+)$, $\Phi_k^- \in \Phi^-$, $\Phi_k^+ \in \Phi^+$ based on Eq. (11)
    Apply destroy operator and repair operator to obtain a new solution $\widetilde{R}$, verify regional connectivity
    Evaluate the partitioning performance $P\left(\widetilde{R}\right)$
    if $random(0, 1) \leq \min\left(1, e^{-\left(P\left(\widetilde{R}\right) - P(R)\right)/T}\right)$ then
        $R \leftarrow \widetilde{R}$
        if $P\left(\widetilde{R}\right) < P(R^*)$ then
            $R^* \leftarrow \widetilde{R}$
    Compute score based on the execution outcome
    Update the number of times elements have been selected in the optional coupling matrix
    Update the average award of operator pairs according to Eq. (12)
    Update regional root after a fixed number of iterations
    $T \leftarrow T \cdot \xi$
return $R^*$
```

A.2. Computational tractability analysis

This experiment analyzes the computational time required to solve the MILP model in Section 2.3 using the solver for different network sizes. The primary aim is to verify the feasibility of obtaining an exact, optimal region-level partition solution and to provide insight into the selection of the solution algorithm.

The real road network from Section 3 is used for the experiment. Subregion adjacency graphs with varying scales are generated by adjusting the internal thresholds in the first-level regionalization model. Given that the number of regions predefined in the second-level regionalization model (i.e., the MILP model) significantly influences the model's solving efficiency, this experiment further analyzes the solution time of the MILP model for different numbers of regions. All other model parameters are consistent with those in Section 3.2. The COIN-OR Branch-and-Cut solver is employed to solve the MILP model, and the computational equipment used is identical to that described in Section 3.

Table A1 presents the solution times for the max-p region and MILP models with varying internal thresholds and numbers of regions. As indicated in the table, the solution time is more substantially impacted by the number of regions than by the size of the subregion adjacency graph (i.e., number of subregions). Specifically, when the number of regions reaches five, the regionalization model for the subregion adjacency graph with 28 nodes becomes challenging to solve within practical time limits (one hour in this experiment). However, as the size of the graph decreases, the computational time for different numbers of regions decreases significantly. Notably, when the subregion adjacency graph contains 17 nodes and the number of regions is five, the MILP model's solution time decreases substantially to 147 s. This suggests that solving the MILP problem using exact algorithms may be impractical when both the number of subregions and regions are large. However, as previously mentioned, in order to define an MFD for perimeter control within each subregion, the subregions must be large enough to maintain the relevance of macroscopic traffic modeling. Therefore, it is reasonable to assume that a region-subregion system constructed for real-world road networks would not involve an excessive number of subregions. In such cases, using exact algorithms in the solver to solve the MILP model ensures optimality while maintaining computational efficiency.

Table A1
Model computational time.

Internal threshold	Number of subregions	Time for max-p region (s)	Number of regions	Time for MILP (s)
10	81	45	3	121
			4	760
			5	>3600
20	42	45	3	40
			4	600
			5	>3600
30	28	49	3	19
			4	277
			5	>3600
40	22	64	3	12
			4	79
			5	1673
50	17	65	3	6
			4	24
			5	147

A.3. MCTS algorithm

Algorithm 2. Monte Carlo Tree Search

```
Input:
    Root state $\mathscr{s}_0$ (current network partitioning)
    Max iterations $I$, Simulation depth $D_S$, Decay factor $\zeta$
    Exploration constant $\omega$, $CV$ threshold $\delta$
Output:
    Optimal action $\mathscr{a}^*$ (subregion movement)
Initialize root node $v_0$ with state $\mathscr{s}_0$
for iteration = 1 to $I$ do
    $v \leftarrow v_0$, $\mathscr{s} \leftarrow \mathscr{s}_0$
    // Selection Phase
    while $v$ is fully expanded and $v$ has children do
        if a random trial (with small probability) occurs then
            Select a child $u$ via a SoftMax policy over estimated rewards
        else
            Select child $u$ using Eq. (32)
        Apply action $\mathscr{a}_u$, update state $\mathscr{s} \leftarrow \widetilde{\mathscr{s}}$
        $v \leftarrow u$
   // Expansion Phase
   if untried actions exist then
        Generate valid actions $\mathscr{A}(\mathscr{s}) = \{(\mathscr{b}, r) | \mathscr{b} \in \mathscr{B}, r \in N(\mathscr{b}, \mathscr{R}), Valid(\mathscr{b}, r, \mathscr{R})\}$
        Calculate move metric $\mathscr{g}_{\mathscr{a}}$
        Select $\mathscr{a}$ from $\mathscr{A}(\mathscr{s})$ with probability $\propto \mathscr{g}_{\mathscr{a}}$
        Create new node $\widetilde{v}$, add to tree
        $v \leftarrow \widetilde{v}$
   // Simulation Phase
   Total reward $\Lambda_t \leftarrow 0$
   for $d_s = 1$ to $D_S$ do
        Select action $\mathscr{a}_s$ from $\mathscr{A}(\mathscr{s})$ using $\varepsilon$-greedy policy
        Apply action, get reward $\Lambda_{d_s}$
        $\Lambda_t = \Lambda_t + \zeta^{d_s - 1} \cdot \Lambda_{d_s}$
        if $CV(r_i) \le \delta, \quad \forall r_i \in \mathscr{R}$ then break
// Backpropagation Phase
   while $v \ne v_0$ do
        Update statistics:
            Visit count $\mathscr{N}(v) \leftarrow \mathscr{N}(v) + 1$
            Cumulative reward $\mathscr{Q}(v) \leftarrow \mathscr{Q}(v) + \Lambda_t$
        $v \leftarrow parent(v)$
// Action Selection
$\mathscr{a}^* \leftarrow \underset{\mathscr{a}}{argmax} \frac{\mathscr{Q}(v_0.children)}{\mathscr{N}(v_0.children)}$
Validate $\mathscr{a}^*$ considering:
    Cluster connectivity preservation
    Tabu list constraints (recently moved subregions)
return $\mathscr{a}^*$
```

A.4. Parameter setting

Table A2
Parameter settings of ANLS.

Parameter	Setting	Parameter	Setting
Max iterations	1000	Destroyed node ratio	0.1
Hierarchical threshold	2	Initial temperature	100
Final temperature	0.1	Trade-off parameter α	0.1

A.5. Demand profile

Taz 0	Taz 1	Taz 2	Taz 3	Taz 4	Taz 5	Taz 6	Taz 7	Taz 8	Taz 9	Taz 10	Taz 11	Taz 12
20	30	30	30	30	30	30	30	30	30	30	30	30
1000	30	30	30	30	30	30	30	30	30	30	30	30
1200	30	30	30	30	30	30	30	30	30	30	30	30
1400	30	30	30	30	30	30	30	30	30	30	30	30
1600	30	30	30	30	30	30	30	30	30	30	30	30
1200	30	30	30	30	30	30	30	30	30	30	30	30
1400	30	30	30	30	30	30	30	30	30	30	30	30
1400	30	30	30	30	30	30	30	30	30	30	30	30

(*continued on next page*)

(*continued*)

Taz 0	Taz 1	Taz 2	Taz 3	Taz 4	Taz 5	Taz 6	Taz 7	Taz 8	Taz 9	Taz 10	Taz 11	Taz 12
1200	30	30	30	30	30	30	30	30	30	30	30	30
1600	30	30	30	30	30	30	30	30	30	30	30	30
1400	30	30	30	30	30	30	30	30	30	30	30	30
1200	30	30	30	30	30	30	30	30	30	30	30	30
1000	30	30	30	30	30	30	30	30	30	30	30	30

Data availability

The data that has been used is confidential.

References

Aalipour, A., Kebriaei, H., Ramezani, M., 2018. Analytical optimal solution of perimeter traffic flow control based on MFD dynamics: a Pontryagin’s maximum principle approach. IEEE Trans. Intell. Transp. Syst. 20 (9), 3224–3234.

Aboudolas, K., Geroliminis, N., 2013. Perimeter and boundary flow control in multi-reservoir heterogeneous networks. Transp. Res. B Methodol. 55, 265–281.

Aimsun, 2024. Aimsun Next 24 User’s Manual, Aimsun Next Version 24.0.0, Barcelona, Spain. Accessed on: April. 16, 2024. [Online].

Ambühl, L., Loder, A., Leclercq, L., Menendez, M., 2021. Disentangling the city traffic rhythms: a longitudinal analysis of MFD patterns over a year. Transp. Res. Part C Emerging Technol. 126, 103065.

Ambühl, L., Loder, A., Zheng, N., Axhausen, K.W., Menendez, M., 2019. Approximative network partitioning for MFDs from stationary sensor data. Transp. Res. Rec. 2673 (6), 94–103.

Ambühl, L., Menendez, M., 2016. Data fusion algorithm for macroscopic fundamental diagram estimation. Transp. Res. Part C Emerging Technol. 71, 184–197.

Assunção, R.M., Neves, M.C., Câmara, G., da Costa Freitas, C., 2007. Efficient regionalization techniques for socio-economic geographical units using minimum spanning trees. Int. J. Geogr. Inf. Sci. 20 (7), 797–811.

Batista, S., Cantelmo, G., Menéndez, M., Antoniou, C., Leclercq, L., 2025. Activity-based user equilibrium considering aggregated traffic dynamics emulated using the Macroscopic Fundamental Diagram. Transp. Res. Part C Emerging Technol. 171, 104980.

Batista, S., Leclercq, L., Menéndez, M., 2021. Dynamic Traffic Assignment for regional networks with traffic-dependent trip lengths and regional paths. Transp. Res. Part C Emerging Technol. 127, 103076.

Batista, S.F., Leclercq, L., 2019. Regional dynamic traffic assignment framework for macroscopic fundamental diagram multi-regions models. Transp. Sci. 53 (6), 1563–1590.

Bellocchi, L., Geroliminis, N., 2020. Unraveling reaction-diffusion-like dynamics in urban congestion propagation: Insights from a large-scale road network. Sci. Rep. 10 (1), 4876.

Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., Colton, S., 2012. A survey of monte carlo tree search methods. IEEE Trans. Comput. Intell. AI Games 4 (1), 1–43.

Buisson, C., Ladier, C., 2009. Exploring the impact of homogeneity of traffic measurements on the existence of macroscopic fundamental diagrams. Transp. Res. Rec. 2124 (1), 127–136.

Chen, C., Geroliminis, N., Zhong, R., 2024a. An iterative adaptive dynamic programming approach for macroscopic fundamental diagram-based perimeter control and route guidance. Transp. Sci. 58 (4), 896–918.

Chen, C., Huang, Y., Lam, W.H., Pan, T., Hsu, S., Sumalee, A., Zhong, R., 2022. Data efficient reinforcement learning and adaptive optimal perimeter control of network traffic dynamics. Transp. Res. Part C Emerging Technol. 142, 103759.

Chen, C., Huang, Y., Zhang, H., Hsu, S.-C., Zhong, R., 2024b. Tracking Perimeter Control for Two-Region Macroscopic Traffic Dynamics: an Adaptive Dynamic programming Approach. In: 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1342–1347.

Coulom, R., 2006. Efficient selectivity and backup operators in Monte-Carlo tree search, International conference on computers and games. Springer 72–83.

Daganzo, C.F., 2007. Urban gridlock: Macroscopic modeling and mitigation approaches. Transp. Res. B Methodol. 41 (1), 49–62.

Daganzo, C.F., Geroliminis, N., 2008. An analytical approximation for the macroscopic fundamental diagram of urban traffic. Transp. Res. B Methodol. 42 (9), 771–781.

Ding, H., Di, Y., Feng, Z., Zhang, W., Zheng, X., Yang, T., 2022. A perimeter control method for a congested urban road network with dynamic and variable ranges. Transp. Res. B Methodol. 155, 160–187.

Ding, H., Wang, L., Zheng, N., Cheng, Z., Zheng, X., Li, J., 2025. A novel hierarchical perimeter control method for road networks considering boundary congestion in a mixed CAV and HV traffic environment. Transp. Res. B Methodol. 195, 103219.

Du, J., Rakha, H., Gayah, V.V., 2016. Deriving macroscopic fundamental diagrams from probe data: issues and proposed solutions. Transp. Res. Part C Emerging Technol. 66, 136–149.

Duncan, L.C., Rasmussen, T.K., Watling, D.P., Nielsen, O.A., 2025. Dynamic multi-region MFD stochastic user equilibrium: Formulation and parameter estimation in a large-scale case study. Transportation Research Part C: Emerging Technologies.

Duque, J.C., Anselin, L., Rey, S.J., 2012. The max-p-regions problem. J. Reg. Sci. 52 (3), 397–419.

El Bukhari, A., Moshahedi, N., Kattan, L., 2024. Optimized locating of sensors for approximating the macroscopic fundamental diagram: a PCA-based approach. Transp. Res. Part C Emerging Technol. 163, 104627.

Fu, H., Chen, S., Chen, K., Kouvelas, A., Geroliminis, N., 2021. Perimeter control and route guidance of multi-region mfd systems with boundary queues using colored petri nets. IEEE Trans. Intell. Transp. Syst. 23 (8), 12977–12999.

Genser, A., Kouvelas, A., 2022. Dynamic optimal congestion pricing in multi-region urban networks by application of a Multi-Layer-Neural network. Transp. Res. Part C Emerging Technol. 134, 103485.

Geroliminis, N., Daganzo, C.F., 2008. Existence of urban-scale macroscopic fundamental diagrams: some experimental findings. Transp. Res. B Methodol. 42 (9), 759–770.

Geroliminis, N., Haddad, J., Ramezani, M., 2012. Optimal perimeter control for two urban regions with macroscopic fundamental diagrams: a model predictive approach. IEEE Trans. Intell. Transp. Syst. 14 (1), 348–359.

Godfrey, J., 1969. The mechanism of a road network. Traffic Eng. Control 11 (7), 323–327.

Gu, Z., Saberi, M., 2019. A bi-partitioning approach to congestion pattern recognition in a congested monocentric city. Transp. Res. Part C Emerging Technol. 109, 305–320.

Guo, D., 2008. Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). Int. J. Geogr. Inf. Sci. 22 (7), 801–823.

Haddad, J., Ramezani, M., Geroliminis, N., 2013. Cooperative traffic control of a mixed network with two urban regions and a freeway. Transp. Res. B Methodol. 54, 17–36.

Haddad, J., Zheng, Z., 2020. Adaptive perimeter control for multi-region accumulation-based models with state delays. Transp. Res. B Methodol. 137, 133–153.

Hamedmoghadam, H., Zheng, N., Li, D., Vu, H.L., 2022. Percolation-based dynamic perimeter control for mitigating congestion propagation in urban road networks. Transp. Res. Part C Emerging Technol. 145, 103922.

Hendel, G., 2022. Adaptive large neighborhood search for mixed integer programming. Math. Program. Comput. 14 (2), 185–221.

Herman, R., Prigogine, I., 1979. A two-fluid approach to town traffic. Science 204 (4389), 148–151.

Hu, C., Tang, J., Li, Z., Wang, Y., Zhao, C., Chen, J., Zhou, H., 2025a. Estimating MFD model parameters from sparse license plate recognition data: the role of path reconstruction and regionalization. Transp. Res. Part C Emerging Technol. 171, 104982.

Hu, C., Tang, J., Wang, Y., Li, Z., Dai, G., 2025b. A flexible road network partitioning framework for traffic management via graph contrastive learning and multi-objective optimization. Comput. Aided Civ. Inf. Eng.

Hu, Z., Ma, W., 2024. Demonstration-guided deep reinforcement learning for coordinated ramp metering and perimeter control in large scale networks. Transp. Res. Part C Emerging Technol. 159, 104461.

Huang, Y., Xiong, J., Sumalee, A., Zheng, N., Lam, W., He, Z., Zhong, R., 2020. A dynamic user equilibrium model for multi-region macroscopic fundamental diagram systems with time-varying delays. Transp. Res. B Methodol. 131, 1–25.

Ji, Y., Geroliminis, N., 2012. On the spatial partitioning of urban transportation networks. Transp. Res. B Methodol. 46 (10), 1639–1656.

Ji, Y., Luo, J., Geroliminis, N., 2014. Empirical observations of congestion propagation and dynamic partitioning with probe data for large-scale systems. Transp. Res. Rec. 2422 (1), 1–11.

Jiang, S., Keyvan-Ekbatani, M., 2023. Hybrid perimeter control with real-time partitions in heterogeneous urban networks: an integration of deep learning and MPC. Transp. Res. Part C Emerging Technol. 154, 104240.

Jiang, S., Keyvan-Ekbatani, M., Ngoduy, D., 2023. Partitioning of urban networks with polycentric congestion pattern for traffic management policies: Identifying protected networks. Comput. Aided Civ. Inf. Eng. 38 (4), 508–527.

Johari, M., Keyvan-Ekbatani, M., Leclercq, L., Ngoduy, D., Mahmassani, H.S., 2021. Macroscopic network-level traffic models: Bridging fifty years of development toward the next era. Transp. Res. Part C Emerging Technol. 131, 103334.

Keyvan-Ekbatani, M., Carlson, R.C., Knoop, V.L., Papageorgiou, M., 2021. Optimizing distribution of metered traffic flow in perimeter control: queue and delay balancing approaches. Control Eng. Pract. 110, 104762.

Keyvan-Ekbatani, M., Papageorgiou, M., Knoop, V.L., 2015. Controller design for gating traffic control in presence of time-delay in urban road networks. Transp. Res. Part C Emerging Technol. 59 (6), 308–322.

Kirkley, A., 2022. Spatial regionalization based on optimal information compression. Communications Physics 5 (1), 249.

Kocsis, L., Szepesvári, C., 2006. Bandit based monte-carlo planning. European Conference on Machine Learning. Springer 282–293.

Leclercq, L., Parzani, C., Knoop, V.L., Amourette, J., Hoogendoorn, S.P., 2015. Macroscopic traffic dynamics with heterogeneous route patterns. Transp. Res. Procedia 7, 631–650.

Li, X., Zhang, X., Qian, X., Zhao, C., Guo, Y., Peeta, S., 2024. Beyond centralization: Non-cooperative perimeter control with extended mean-field reinforcement learning in urban road networks. Transp. Res. B Methodol. 186, 103016.

Li, Y., Mohajerpoor, R., Ramezani, M., 2021. Perimeter control with real-time location-varying cordon. Transp. Res. B Methodol. 150, 101–120.

Liu, H., Gayah, V.V., 2024. N-MP: A network-state-based Max Pressure algorithm incorporating regional perimeter control. Transportation Research Part C: Emerging Technologies, 104725.

Loder, A., Bliemer, M.C., Axhausen, K.W., 2022. Optimal pricing and investment in a multi-modal city-introducing a macroscopic network design problem based on the MFD. Transp. Res. A Policy Pract. 156, 113–132.

Lopez, C., Krishnakumari, P., Leclercq, L., Chiabaut, N., Van Lint, H., 2017a. Spatiotemporal partitioning of transportation network using travel time data. Transp. Res. Rec. 2623 (1), 98–107.

Lopez, C., Leclercq, L., Krishnakumari, P., Chiabaut, N., Van Lint, H., 2017b. Revealing the day-to-day regularity of urban congestion patterns with 3D speed maps. Sci. Rep. 7 (1), 14029.

Lougee-Heimer, R., 2003. The common optimization interface for operations research: promoting open-source software in the operations research community. IBM J. Res. Dev. 47 (1), 57–66.

Ma, J., Wu, F., 2023. Learning to coordinate traffic signals with adaptive network partition. IEEE Trans. Intell. Transp. Syst. 25 (1), 263–274.

Mahmassani, H., Williams, J.C., Herman, R., 1987. Performance of urban traffic networks, Proceedings of the 10th International Symposium on Transportation and Traffic Theory. Elsevier Amsterdam, The Netherlands, pp. 1-20.

Mahmassani, H.S., Saberi, M., Zockaie, A., 2013. Urban network gridlock: theory, characteristics, and dynamics. Transp. Res. Part C Emerging Technol. 36, 480–497.

Mahmassani, H.S., Williams, J.C., Herman, R., 1984. Investigation of network-level traffic flow relationships: some simulation results. Transp. Res. Rec. 971, 121–130.

Mara, S.T.W., Norcahyo, R., Jodiawan, P., Lusiantoro, L., Rifai, A.P., 2022. A survey of adaptive large neighborhood search algorithms and applications. Comput. Oper. Res. 146, 105903.

Mariotte, G., Leclercq, L., 2019. Flow exchanges in multi-reservoir systems with spillbacks. Transp. Res. B Methodol. 122, 327–349.

Mariotte, G., Leclercq, L., Batista, S., Krug, J., Paipuri, M., 2020. Calibration and validation of multi-reservoir MFD models: a case study in Lyon. Transp. Res. B Methodol. 136, 62–86.

Mitchell, S., OSullivan, M., Dunning, I., 2011. Pulp: a linear programming toolkit for python. The University of Auckland, Auckland, New Zealand 65, 25.

Moshahedi, N., Kattan, L., 2022. A macroscopic dynamic network loading model using variational theory in a connected and autonomous vehicle environment. Transp. Res. Part C Emerging Technol. 145, 103911.

Mousavizadeh, O., Keyvan-Ekbatani, M., 2024. On the important features for a well-shaped reduced network MFD estimation during network loading and recovery. Transp. Res. Part C Emerging Technol. 161, 104539.

Niu, X.-J., Zhao, X.-M., Xie, D.-F., Liu, F., Bi, J., Lu, C., 2022. Impact of large-scale activities on macroscopic fundamental diagram: Field data analysis and modeling. Transp. Res. A Policy Pract. 161, 241–268.

Paipuri, M., Leclercq, L., Krug, J., 2019. Validation of macroscopic fundamental diagrams-based models with microscopic simulations on real networks: Importance of production hysteresis and trip lengths estimation. Transp. Res. Rec. 2673 (5), 478–492.

Paipuri, M., Xu, Y., González, M.C., Leclercq, L., 2020. Estimating MFDs, trip lengths and path flow distributions in a multi-region setting using mobile phone data. Transp. Res. Part C Emerging Technol. 118, 102709.

Pascale, A., Mavroeidis, D., Lam, H.T., 2015. Spatiotemporal clustering of urban networks: real case scenario in London. Transp. Res. Rec. 2491 (1), 81–89.

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Comput. Oper. Res. 34 (8), 2403–2435.

Ramezani, M., Haddad, J., Geroliminis, N., 2015. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. Transp. Res. B Methodol. 74, 1–19.

Ropke, S., Pisinger, D., 2006a. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. 40 (4), 455–472.

Ropke, S., Pisinger, D., 2006b. A unified heuristic for a large class of vehicle routing problems with backhauls. Eur. J. Oper. Res. 171 (3), 750–775.

Saedi, R., Saeedmanesh, M., Zockaie, A., Saberi, M., Geroliminis, N., Mahmassani, H.S., 2020. Estimating network travel time reliability with network partitioning. Transp. Res. Part C Emerging Technol. 112, 46–61.

Saeedmanesh, M., Geroliminis, N., 2016. Clustering of heterogeneous networks with directional flows based on "Snake" similarities. Transp. Res. B Methodol. 91, 250–269.

Saeedmanesh, M., Geroliminis, N., 2017. Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. Transp. Res. B Methodol. 105, 193–211.

Saffari, E., Yildirimoglu, M., Hickman, M., 2020. A methodology for identifying critical links and estimating macroscopic fundamental diagram in large-scale urban networks. Transp. Res. Part C Emerging Technol. 119, 102743.

Saffari, E., Yildirimoglu, M., Hickman, M., 2023. Estimation of macroscopic fundamental diagram solely from probe vehicle trajectories with an unknown penetration rate. IEEE Trans. Intell. Transp. Syst.

Shaw, P., 1998. Using constraint programming and local search methods to solve vehicle routing problems, International conference on principles and practice of constraint programming. Springer 417–431.

She, B., Duque, J.C., Ye, X., 2016. The network-max-p-regions model. Int. J. Geogr. Inf. Sci. 31 (5), 962–981.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 888–905.

Shim, J., Yeo, J., Lee, S., Hamdar, S.H., Jang, K., 2019. Empirical evaluation of influential factors on bifurcation in macroscopic fundamental diagrams. Transp. Res. Part C Emerging Technol. 102, 509–520.

Sirmatel, I.I., Geroliminis, N., 2017. Economic model predictive control of large-scale urban road networks via perimeter control and regional route guidance. IEEE Trans. Intell. Transp. Syst. 19 (4), 1112–1121.

Świechowski, M., Godlewski, K., Sawicki, B., Mańdziuk, J., 2023. Monte Carlo tree search: a review of recent modifications and applications. Artif. Intell. Rev. 56 (3), 2497–2562.

Tsitsokas, D., Kouvelas, A., Geroliminis, N., 2023. Two-layer adaptive signal control framework for large-scale dynamically-congested networks: combining efficient Max pressure with Perimeter Control. Transp. Res. Part C Emerging Technol. 152, 104128.

Wei, R., Rey, S., Knaap, E., 2020. Efficient regionalization for spatially explicit neighborhood delineation. Int. J. Geogr. Inf. Sci. 35 (1), 135–151.

Yan, F., Zhang, M., Shi, Z., 2021. Dynamic partitioning of urban traffic network sub-regions with spatiotemporal evolution of traffic flow. Nonlinear Dyn. 105 (1), 911–929.

Yildirimoglu, M., Geroliminis, N., 2014. Approximating dynamic equilibrium conditions with macroscopic fundamental diagrams. Transp. Res. B Methodol. 70, 186–200.

Yildirimoglu, M., Ramezani, M., Geroliminis, N., 2015. Equilibrium analysis and route guidance in large-scale networks with MFD dynamics. Transp. Res. Part C Emerging Technol. 59, 404–420.

Yildirimoglu, M., Sirmatel, I.I., Geroliminis, N., 2018. Hierarchical control of heterogeneous large-scale urban road networks via path assignment and regional route guidance. Transp. Res. B Methodol. 118, 106–123.

Yu, J., Laharotte, P.-A., Han, Y., Ma, W., Leclercq, L., 2025. Perimeter control with heterogeneous metering rates for cordon signals: a physics-regularized multi-agent reinforcement learning approach. Transp. Res. Part C Emerging Technol. 171, 104944.

Zhang, L., Yuan, Z., Yang, L., Liu, Z., 2020. Recent developments in traffic flow modeling using macroscopic fundamental diagram. Transp. Rev. 40 (4), 529–550.

Zheng, N., Geroliminis, N., 2016. Modeling and optimization of multimodal urban networks with limited parking and dynamic pricing. Transp. Res. B Methodol. 83, 36–58.

Zhou, D., Gayah, V.V., 2021. Model-free perimeter metering control for two-region urban networks using deep reinforcement learning. Transp. Res. Part C Emerging Technol. 124, 102949.

Zhou, D., Gayah, V.V., 2023. Scalable multi-region perimeter metering control for urban networks: a multi-agent deep reinforcement learning approach. Transp. Res. Part C Emerging Technol. 148, 104033.

Zockaie, A., Saberi, M., Saedi, R., 2018. A resource allocation problem to estimate network fundamental diagram in heterogeneous networks: Optimal locating of fixed measurement points and sampling of probe trajectories. Transp. Res. Part C Emerging Technol. 86, 245–262.