Transportation Research Part C 164 (2024) 104671

Contents lists available at ScienceDirect

Transportation Research Part C

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

Computational graph-based mathematical programming reformulation for integrated demand and supply models[☆]

Taehooie Kim[a,*], Jiawei Lu[b], Ram M. Pendyala[c], Xuesong Simon Zhou[c]

[a] UrbanSim Inc., 221 1st Ave W, Seattle, WA 98119, USA
[b] Georgia Institute of Technology, Atlanta, GA 30332, USA
[c] Arizona State University, School of Sustainable Engineering and the Built Environment, 660 S. College Avenue, Tempe, AZ 85287-3005, USA

ARTICLE INFO

Keywords:
Computational graphs (CGs)
Automatic differentiation (AD)
Lagrangian relaxation (LR)
Alternating direction method of multipliers (ADMM)
And the integration of travel demand and transportation network

ABSTRACT

As transportation systems grow in complexity, analysts need sophisticated tools to understand travelers' decision-making and effectively quantify the benefits of the proposed strategies. The transportation community has developed integrated demand–supply models to capture the emerging interactive nature of transportation systems, serve diverse planning needs, and encompass broader solution possibilities. Recently, utilizing advances in Machine Learning (ML) techniques, researchers have also recognized the need for different computational models capable of fusing/analyzing different data sources. Inspired by this momentum, this study proposes a new modeling framework to analytically bridge travel demand components and network assignment models with machine learning algorithms. Specifically, to establish a consistent representation of such aspects between separate system models, we introduce several important mathematical programming reformulation techniques—variable splitting and augmented Lagrangian relaxation—to construct a computationally tractable nonlinear unconstrained optimization program. Furthermore, to find equilibrium states, we apply automatic differentiation (AD) to compute the gradients of decision variables in a layered structure with the proposed model represented based on computational graphs (CGs) and solve the proposed formulation through the alternating direction method of multipliers (ADMM) as a dual decomposition method. Thus, this reformulated model offers a theoretically consistent framework to express the gap between the demand and supply components and lays the computational foundation for utilizing a new generation of numerically reliable optimization solvers. Using a small example network and the Chicago sketch transportation network, we examined the convergency/consistency measures of this new differentiable programming-based optimization structure and demonstrated the computational efficiency of the proposed integrated transportation demand and supply models.

1. Introduction

Faced with multiple data sources including travel surveys, mobile phone data records, GPS, or sensor data, and the increased bulk of the datasets, the transportation community has actively applied advanced numerical and computational techniques to effectively

[☆] This article belongs to the Virtual Special Issue on "special issue full title".
[*] Corresponding author.
E-mail addresses: taehooie@urbansim.com (T. Kim), jlu486@gatech.edu (J. Lu), ram.pendyala@asu.edu (R.M. Pendyala), xzhou74@asu.edu (X.S. Zhou).

https://doi.org/10.1016/j.trc.2024.104671
Received 24 September 2023; Received in revised form 18 April 2024; Accepted 17 May 2024
Available online 31 May 2024

analyze the massive data sources and detect unseen trends, helping alleviate traffic issues. Along this line, diverse ML algorithms/methods and deep learning (DL) architectures have been constructed to understand real-time management information for large fleets (Safikhani et al., 2017; Lin et al., 2018), find unobserved patterns in data sets (Ashraf et al., 2020), and enhance travel demand models (Wong and Farooq, 2019; Sifringer et al., 2020; Kim et al., 2020; Wang et al., 2020a, 2020b; Zhao et al., 2020; Yan et al., 2020; Liu et al., 2021). The ML-oriented application in the transportation field has demonstrated excellent computational capabilities in enhancing predictive power, analyzing unobserved complex patterns, and dealing with large-scale datasets efficiently, thereby allowing planners and agencies to model patterns of emerging disruptions. More recently, scholars have been actively reviewing both fields to differentiate the role of each approach (i.e., data-driven approaches and theory-based models) and enhancing domestic knowledge-based structure through automatic learning (Wang et al., 2021; Van Cranenburgh et al., 2022; Han et al., 2022).

Based on the recent computational advance of ML applications, this paper aims to develop a theoretically sound, and computationally efficient framework to couple transportation demand and network models to quantify the interaction and impact of a wide range of congestion mitigation strategies. In particular, this research adopts a new generation of computational methods and optimization paradigms from the machine learning community—computational graphs (CGs) and backpropagation—to uniquely capture the layered modeling structure in the integrated demand–supply model. Furthermore, this analytically driven formulation and resulting computational architecture would extend the ability to ensure a high degree of consistency between transportation demand and supply models, while maintaining sufficient descriptive capability for the interactive transportation system. This paper is also intended to address pertinent questions asked by transportation decision-makers: how to quantify the nature of complex transportation system interactions, specifically human mobility pattern changes and their impact on the built infrastructures. With the utilization of high-performance computing (HPC), we resolve the complex mathematical formulation with optimal solutions. In the subsequent section, we address prior work on how the community has developed integrated transportation models.

1.1. Integration of transportation demand and network models

With the increasing complexity of planning transportation infrastructure in urban areas, the transportation modeling community has dedicated significant efforts to develop integrated transportation demand and network models. Demand components are generally modeled using trip-based or activity-based approaches, and network model structures are represented through static user equilibrium or dynamic traffic assignment (Boyles et al., 2022). A feedback loop structure is constructed from the lower-level route assignment to the upper-level trip generation, destination choice, and mode choice layers to reach a consistent representation for travel time measures across different steps. Recent efforts in the last 20 years have been actively devoted to integrating tour-based modeling approaches on the demand side and dynamic traffic assignment models on the supply side to capture the full spectrum of traveler dynamics and resulting time-varying traffic congestion (Esser and Nagel, 2001; Lam and Huang, 2003; Raney et al., 2003).

As many implementations adopt a microsimulation approach to linking transportation demand and network models, various convergence criteria are defined based on a fixed-point formulation and software-oriented system coupling calls for iterative solution methods using computed travel time skimmed profiles (Lin et al., 2008; Hao et al., 2010; Pendyala et al., 2012). For instance, Lin et al. (2008) proposed a fixed-point formulation within a variational inequality framework with a solution based on the method of successive averages (MSA). To further integrate travel demand and supply models with long-term land-use evolution, Pendyala et al. (2012) developed a tightly coupled framework across three open-source packages, with various iterative processes accommodating different time-updating resolutions, such as En-route real-time decisions, post-trip learning within a day, and year-by-year long-term

Table 1
Recent Studies for Coupling Demand-Supply Models and Corresponding Solution Algorithms.

Publication	Systematic flows	Formulations for interactive systems	Gradient Computation	Step sizes for finding optimal flow patterns and travel costs	Solution algorithms for the demand and supply variables
Florian et al. (2002)	flow → cost	Variational inequality (VI)	No	Predetermined decreasing sequence	External Block Gauss-Seidel decomposition with MSA
Boyce et al. (2008)	flow → cost	Fixed point (FP)	No	Predetermined decreasing sequence	Averaging with fixed weights and MSA
Lin et al. (2008)	flow → cost	Variational inequality (VI)	No	Predetermined decreasing sequence	External sequential process with MSA
Zhou et al. (2008) and Lu et al. (2009)	flow ↔ cost	Gap function	No	Predetermined decreasing sequence	Internal circular process with MSA
Zhou et al. (2009)	flow → cost	Variational inequality (VI)	No	Rule-based algorithm	External sequential process with the self-adaptive GLP projection
Cantarella et al. (2015)	flow ↔ cost	Fixed point (FP)	No	Predetermined decreasing sequence	Internal circular process with MSA
Ryu et al. (2017)	flow → cost	Mathematical program (MP)	No	Predetermined decreasing sequence	External gradient projection (GP)
This paper	Behavior choice ↔ flow ↔ cost	Gap function-based reformulation using ADMM	Yes	Analytic gradients via automatic differentiation (AD)	Internal circular process with gradient-based numerical optimization (e.g., BFGS)

decision adjustment. There are a number of integrated model efforts between activity based model and dynamic traffic assignments, to name a few, Konduri (2012), Flötteröd et al. (2012), Kang (2013), Habib et al. (2013), Zhang et al. (2018), Xu et al. (2017), Halat et al. (2017), Xiong et al. (2018), Zhu et al. (2018), Liu et al. (2018), Cantelmo and Viti (2019), Wang et al. (2020a, 2020b), and Najmi et al. (2020).

Many other related studies have also implemented a similar system integration structure with a feedback loop system between underlying subcomponents. These research efforts with different defining systematic structures can be examined according to three aspects (Boyce et al., 2004): mathematical formulations, solution methods, and convergence criteria (Zhou et al., 2009; Yao et al., 2014; Verbas et al., 2016; Chu, 2018; Patil et al., 2021). In integration modeling studies specifically developed for combined modal split and traffic assignment problems (CMSTA), researchers have developed different model formulations and corresponding solution algorithms to find sets of optimal path flow patterns and generalized travel costs under different criteria (e.g., user equilibrium and stochastic behavior) to quantify system behavior among system aspects and traveler choice. Recent representative studies linking travel demand and transportation supply are systematically reviewed and summarized in Table 1.

Table 1 indicates the systematic directions of coupling demand and supply, formulations for interactive modeling systems, presence of gradient information, step sizes used in updating the defined formulations, and solution algorithms. Most existing studies on integrated demand–supply models have shown unidirectional or bidirectional systematic flows when finding optimal path flow patterns and generalized travel costs. In addition, different solution algorithms (e.g., MSA, gradient projection (GP), or self-adaptive gradient projection) have been proposed with predetermined step sizes or heuristic-based step sizes to find the optimal solutions in interactive modeling systems built by VI, FP, or MP. In comparison, this study focuses on formulating an analytically defined comprehensive framework to theoretically map behavior choice, path flow patterns, and travel costs, and solve the optimization problem using analytic gradients obtained by automatic differentiation (AD).

1.2. Contributions: Analytical Gradient-Based optimization and computational algorithm

While the existing literature covers a wide variety of mathematical frameworks and solutions for coupling travel demand and supply models, two long-standing challenges from a mathematical modeling perspective still remain in terms of efficiently evaluating and propagating derivatives of variables defined in different components and the recognition of two essential modeling structures: (i) hierarchical layered processes and (ii) complex feedback loops across different layers. Our recent work used a CG-based approach to describe various modeling elements; however, there are still two specific methodological issues to be addressed to fully utilize the new generation of optimization solvers (Wu et al., 2016; Kim et al., 2021).

- First, if a hierarchical system commonly expressed by a fixed-point approach, as shown in Appendix A, is computed by a feedforward approach, which calculates the gradients of input variables and sends their information towards outputs, then the computational performance is significantly dependent on the number of variables to be estimated and optimized, eventually resulting in exponential computing time (Olah, 2015). From a transportation modeling perspective, a sequential structure layered by origin, origin–destination, and numerous paths and their link information includes different numbers of n variables in this four-layered architecture. In this case, the number of possible paths for solving the decision variables is n^4. When dealing with a larger network and a set of variables n, *feedforward computation* leads to combinatorial explosion (Olah, 2015).
- Second, a theoretical modeling framework for formulating an integrated supply–demand model is needed. In particular, gap function-based methods to find *equivalent states* for the two components–transportation demand and network assignment–are required to tightly link transportation demand and supply and measure the extent of inconsistency (Lu et al., 2009).

Motivated by these two key issues, i.e., *feedforward computation* and *equivalent states*, this research focuses on establishing a novel framework for minimizing the inconsistency gap of the submodels and enhancing convergence for interactively layered structures. In particular, the concept of backpropagation, a core algorithm for training machine learning models (LeCun et al., 2015), has recently been applied in optimizing/estimating conventional transportation planning models (Wu et al., 2018; Kim et al., 2021; Lu and Zhou, 2023). According to LeCun et al. (2015) and Olah (2015), the backpropagation algorithm (reverse-mode differentiation) enables a reduction in the computing time to obtain gradients. For instance, the computational complexity of finding optimality in a four-layered structure with n variables can be $4n$, thereby decreasing the cost of calculation and solving the *feedforward computation* issue.

To execute this algorithm, the integrated modeling framework is reformulated using a graph-oriented programming language available to derive the gradients of the decision variables using automatic differentiation (AD), where the gradients of the given functions are computed based on the chain rule. We provide detailed explanations in Appendix B to describe the inherent connections between the chain rule and AD systematically. Details of the advantages of this algorithm can be found in studies by Wright and Nocedal (1999) and Baydin et al. (2017). Furthermore, to achieve a top-down process and discover *equivalent states*, a new constraint for tightly bridging demand/supply components is identified using the variable splitting method (Ortuzar and Willumsen, 2001). The objective function is formulated by augmented Lagrangian relaxation with computational graphs (CGs) to find the optimal values for the network equilibrium states and linking constraints.

This research highlights the benefits of the automated calculation of gradients in a complex and nonlinear structure with multiple layers of composite functions, with the iterative computing of step size handled through underlying solvers, namely TensorFlow (Abadi et al., 2016). Thus, we shift our focus to model reformulation to comprehensively incorporate network flow, utility, and choice probability at different choice dimensions while still taking advantage of emerging computing architecture. To demonstrate the unconstrained nonlinear optimization program, a simple case study and a real network (Chicago Transportation Network) are examined. In this study, we apply the alternating direction method of multipliers (ADMM) as a dual decomposition-based problem–solution paradigm. The survey studied by Boyd et al. (2011) examines the wide use of ADMM in many fields. Several studies in the field of convex programming (Ruszczyński, 1989; Eckstein and Bertsekas, 1992) have systematically conducted convergence analysis and developed many related theoretical aspects, such as distributed optimization with multiple agents (Nedic and Ozdaglar, 2009). Recent ADMM applications in the field of transportation have also been reported. Boland et al. (2018) presented a stochastic form of augmented Lagrangian based on decomposition, and Yao et al. (2019) solved the vehicle routing problem with time windows using the algorithm.

The remainder of this paper is organized as follows. Section 2 presents a problem statement. Section 3 describes the formulation of variational inequality for the DUE and fixed-point equations to sequentially link each interactive variable used in the model. The subsequent section presents the solution algorithm (variable splitting and ADMM applied in reverse-mode differentiation (AD). In Section 5, the performance of the integrated transportation model is evaluated using traffic network examples. Finally, Section 6 summarizes our contributions and extends the current version.

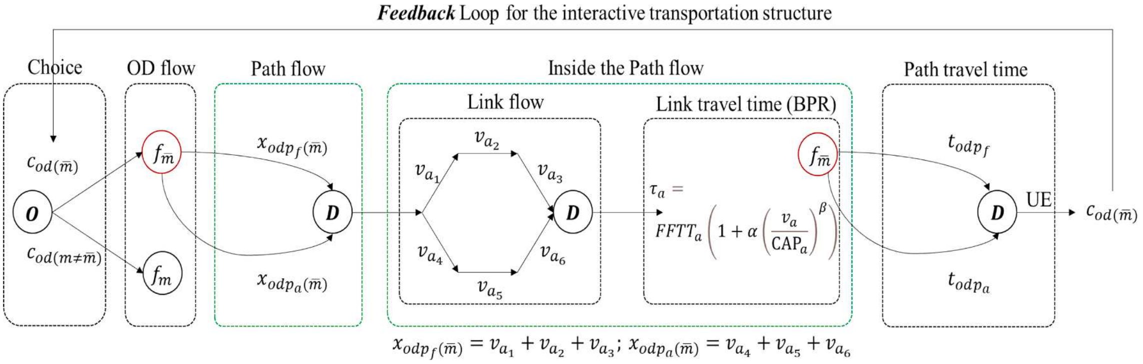
2. Problem statement and conceptual reformulation framework

2.1. Integrated Demand-Supply framework

The deep integration of travel demand and network models is of great interest, and interested readers refer to a recent effort along this line as part of the SHRP II C10 project (Smith et al., 2018). Typically, the demand-side component shares a list of individual trips to the supply-side model, while the aggregated level of service (LOS), in the form of a travel time matrix (i.e., skim) for all possible trips across OD pairs, is fed back from the supply side model to the demand side. Through multiple iterations of the flow adjustment, the ultimate goal is to reduce the inconsistent time-of-day outcomes between the experienced travel time from the demand side and the expected travel time from the supply side. Theoretically, the linkage between different components should be clearly defined within the temporal equilibrium on the demand side and behavioral considerations on the supply side. Without loss of generality, this study aims to find a stable trip demand pattern and equilibrated path flows that minimize system-wide inconsistency measures.

Mathematically, given a transportation network with a set of nodes N, and a set of links A, the problem aims to find a consistent traveler behavior and network performance solution, subject to several demand and supply constraints. In the demand-side network, nodes are designated as activity locations of traffic analysis zones for the high-level traveler behavioral model, with the modeling links indicating the direction of the choice behavior. In the supply side network, the system performance associated with the link traveling costs and path flows is computed on a given physical road or multimodal network. The generalized/travel costs between OD pairs are selected as key consistency/convergence criteria in the demand–supply integration under consideration.

The conceptual model structure is shown in Fig. 1, which denotes a layered system with a set of physical nodes ($n \in N$), a set of physical links $l \in A$, and a set of links related to mode choice behaviors. This sequential modeling data flow starts from origin node $\boldsymbol{o}$. Based on the scale of generalized costs ($c_{od(\overline{m})}$ and $c_{od(m \neq \overline{m})}$), $f_{od(\overline{m})}$ denotes the number of trips selected by trip makers with alternative mode $\overline{m}$. The distribution of the path volumes can also be quantified in $x_{odp_f(\overline{m})}$ and $x_{odp_a(\overline{m})}$, path flows on the freeway corridor f, and arterial corridor a. Inside the path flows, the link flows are mapped by an incidence matrix, and the Bureau of Public Roads (BPR) function is applied to compute the link travel times. t_{odp_f} and t_{odp_a} are the travel time variables corresponding to each path alternative. The travel cost $c_{od(\overline{m})}$ for a certain OD pair should be computed to satisfy the DUE condition. Finally, to minimize the gap of the shared variables $c_{od(\overline{m})}$ between the demand and supply sides, the feedback loop system is activated, leading to a tightly coupled model system. This simplified connectivity aims to demonstrate a parsimonious structure that bridges behavioral elements and network performance elements. According to Guarda et al., (2024), the probabilistic assignment approach helps increases stability of processing network flow estimation. On the one hand, this approach was inspired by the simplification of the modeling structure using DUE, which chooses



Feedback Loop for the interactive transportation structure
Choice
OD flow
Path flow
Inside the Path flow
Link flow
Link travel time (BPR)
Path travel time
$c_{od(\overline{m})}$
O
$c_{od(m \neq \overline{m})}$
$f_{\overline{m}}$
f_m
$x_{odp_f(\overline{m})}$
$x_{odp_a(\overline{m})}$
D
v_{a_1}
v_{a_2}
v_{a_3}
v_{a_4}
v_{a_5}
v_{a_6}
D
$\tau_a = FFTT_a\left(1 + \alpha\left(\frac{v_a}{\mathrm{CAP}_a}\right)^{\beta}\right)$
$f_{\overline{m}}$
t_{odp_f}
t_{odp_a}
D
UE
$c_{od(\overline{m})}$
$x_{odp_f(\overline{m})} = v_{a_1} + v_{a_2} + v_{a_3}$; $x_{odp_a(\overline{m})} = v_{a_4} + v_{a_5} + v_{a_6}$

Fig 1. Conceptual Illustration of Integrated Supply-Demand System in a Simple Network.

the shortest or least costly path to destination. We assumed the simplified computing process could help improve computational efficiency, especially for large-scale networks.

In the above example, there are two essential decision variables: generalized/travel costs $c_{od(\overline{m})}$ and path flow $x_{odp_f(\overline{m})}$ and $x_{odp_a(\overline{m})}$, depending on the number of available paths and OD pairs. It should be noted that a set of composite functions are available to compute the travel cost $c_{od(\overline{m})}$, jointly determined by the choice and network models. Specifically, t_{odp_f} and t_{odp_a} are determined by the volume delay, link performance function, and path-link incidence matrix. The generic formulation of the simplified demand–supply integration program is provided in the following subsection. The indices, variables, and parameters used in the formulation are presented in Table 2. Before discussing the equations for the demand and supply sides, we would like to first introduce a variable splitting method to reformulate the feedback loop system.

2.2. *Variable splitting method for enabling problem decomposition of feedback loops*

In general, successive feedback loops between different elements are used to model mutual causal interactions in transportation networks and ensure equilibrium states (Lin et al., 2008; Pendyala et al., 2012; Chu, 2018). Specifically, the potential travel demand is estimated by econometric models and sent to network models to measure network performance, such as level of service (LOS). Then, the information returns to the demand side to regenerate potential trips and their attributes. This iterative procedure was performed continuously until the integrated models met the convergence criterion. From the perspective of problem decomposition, we highlight the need to mathematically **"decouple"** common variables (i.e., costs) used in both sub-problem models (demand and supply).

A key principle in many large-scale system optimization applications is to first decompose the constraints into "easy" vs. "difficult" constraints and then dualize the "difficult" constraints to enable the resulting components or subproblems can be efficiently solved (Fisher et al., 1997). In addition, variable splitting is another important problem decomposition method, which splits the common variables and creates an additional coupling constraint between "duplicated" variables, and then dualizes the newly introduced coupling constraint in the objective function to generate two "easy-to-solve" subproblems (Guignard, 2003). In Chapter 7 of this study, the detailed concept of variable splitting is explained. Mathematically, we consider a general mathematical function to link the two demand and supply subsystems with essential constraints. The integration optimization problem can be described as $\min_{x}\{I(x)|Ax \leq b, Cx \leq d, x \in X\}$, where $I(x)$ indicates a function for subsystem integration, x is a vector of demand and supply variables that require finding optimal solutions, and the constraints, $Ax \leq b$ and $Cx \leq d$, are associated with the demand and supply sides, respectively. Based on the concept of the variable splitting method, we can decompose a part of the common cost variable x used in both subsystems/constraints as follows: $x \rightarrow x_d$ and x_s, which leads to $\min_{x_d, x_s}\{I(x)|Ax_d \leq b, Cx_s \leq d, x_d \in X \text{and} \, x_s \in X, x_d = x_s\}$. With the conceptual understanding, the proposed decomposition-based approach is illustrated in Appendix C.

Table 2
Notation.

Network information $G = (N, A)$ and indices:	
T_o	Total demand on origin node o
TD_{od}	Total demand from origin node o to destination node d
o	Subscript for an origin node,$\boldsymbol{o} \in \boldsymbol{N}$
d	Subscript for a destination node,$\boldsymbol{d} \in \boldsymbol{N}$
p	Subscript for a path,$p \in P(o, d)$
$\overline{p}$	Subscript for a path to represent the last path variable of a set of path variables $p \neq \overline{p}$
(m)	Alternative modes m (a set of alternative modes); $m = \overline{m}$ (auto mode) and $m \neq \overline{m}$ (transit)
CAP_a	Link capacity on link a
a	Index for a link,$a = 1, 2, \cdots, \lvert A \rvert$
(l)	Superscript for a layer index
Input parameters/values:	
θ	Dispersion parameter
λ, ρ	Lagrange multiplier and penalty term
δ_{odpa}	Incidence matrix for the linkage between path layers and link layers
α, β	The BPR function parameters (e.g., $\alpha = 0.15$ and $\beta = 4$)
$FFTT_a$	Free flow travel time on link a
Decision and intermediate variables:	
$f_{od(\overline{m})}$	Number of trips from node o to node d by mode $\overline{m}$
$P(o, d)$	Set of all feasible paths for a given pair (o, d)
$x_{odp(m)}$	Number of trips from node o to node d in path p by mode m
$x_{odp(\overline{m})}$	Number of trips from node o to node d in path p by mode $\overline{m}$
t_{odp}	Path travel time cost from node o to node d in path p
$c_{od(m)}$	Common variable for supply side and demand side in a given *od* pair with mode m
$c^s_{od(m)}$	Least travel cost, min$t$ in a given *od* pair (supply side) with mode m
$c^d_{od(m)}$	Generalized cost in selecting alternative modes (demand side) with mode m
v_a	Link flows on link a
τ_a	Link travel time on link a

2.3. Iterative method based on dual Decomposition: Alternating direction method of multipliers (ADMM)

With an understanding of the decomposition and dualization of subsystem integration, we introduce the background of the alternating direction method of multipliers (ADMM) algorithm to efficiently solve defined nonlinear continuous optimization problems (Boyd et al., 2011). This approach enables us to decompose an integrated transportation system into travel demand and network assignment components and dualize the constraints of behavioral representation and network flow conservation. To relax the hard constraints, the proposed objective function was augmented by quadratic penalty functions. Solving each subproblem (i.e., demand and supply) iteratively, we update the optimal variables and achieve convergence between the travel demand and network assignment.

In general, ADMM is a combination of the augmented Lagrangian relaxation and block coordinate descent methods, where the algorithm can relax hard constraints and ensure the decomposition of our subproblem (Yao et al., 2019). For instance, consider the problem statement illustrated in Fig. 1. There are two vector variables to be optimized (generalized traveling costs, $c_{od(\overline{m})}$, on the demand side; path flows, $x_{odp(\overline{m})}$, on the supply side). Using the generalized functional formulations of each subsystem, the objective function can be expressed as

$$\min Z = \boldsymbol{D}(c^{d}_{od(\overline{m})}) + \boldsymbol{S}(x^{s}_{odp(\overline{m})}) \tag{1}$$

Subject to

$$A^{d}\boldsymbol{D}(c^{d}_{od(\overline{m})}) = B^{\boldsymbol{d}} \tag{2a}$$

$$C^{s}\boldsymbol{S}(x^{s}_{odp(\overline{m})}) = D^{s} \tag{2b}$$

$$c^{d}_{od(\overline{m})} = c^{s}_{od(\overline{m})} \tag{2c}$$

where Eq. (2a), Eq. (2b), and Eq. (2c) indicate demand, supply, and linkage constraints, respectively. $\boldsymbol{D}$ and $\boldsymbol{S}$ represent generalized functional expressions for demand and supply, respectively. In addition, A^{d}, B^{d}, C^{s}, and D^{s} denote the defined parameters to be calibrated. The superscripts, d and s classify the parameters as demand-related and supply-related, respectively.

By implementing the ADMM-based dual decomposition method, the given constraints of the subsystem components are dualized into the objective function with Lagrangian multipliers λ and penalty parameters $\boldsymbol{\rho}$:

$$\min L_{\rho} = \boldsymbol{D}(c^{d}_{od(\overline{m})}) + \boldsymbol{S}(x^{s}_{odp(\overline{m})}) + \lambda^{d}\left(A^{d}\boldsymbol{D} - B^{d}\right) + \frac{\rho^{d}}{2}\|A^{d}\boldsymbol{D} - B^{d}\|_2^2 + \lambda^{s}(C^{s}\boldsymbol{S} - D^{s}) + \frac{\rho^{s}}{2}\|C^{s}\boldsymbol{S} - D^{s}\|_2^2 + \lambda^{l}\left(c^{d}_{od(\overline{m})} - c^{s}_{od(\overline{m})}\right) + \frac{\rho^{l}}{2}\|c^{d}_{od(\overline{m})} - c^{s}_{od(\overline{m})}\|_2^2 \tag{3}$$

According to the iterative update scheme of ADMM, the variables shown on each side can be sequentially solved with fixed variables. It is worth noting that the ADMM algorithm is similar to a fixed-point iteration method. However, the key difference is that this solving algorithm deals with penalization in the given objective function, leading to a mathematically formulated linkage for the interactive transportation system and enhancing the convergence of the coupled model structure.

Eq. (4)–(8) illustrate the iterative solving process of the ADMM-based solution framework while updating the multipliers.

$$c^{d(n+1)}_{od(\overline{m})} := \text{argmin}_{\boldsymbol{D}} L_{\rho}\left(\boldsymbol{D}\left(c^{d}_{od(\overline{m})}\right), \boldsymbol{S}(x^{s(n)}_{odp(\overline{m})})\right) \tag{4}$$

$$x^{s(n+1)}_{odp(\overline{m})} := \text{argmin}_{\boldsymbol{S}} L_{\rho}\left(\boldsymbol{D}\left(c^{d(n+1)}_{od(\overline{m})}\right), \boldsymbol{S}\left(x^{s}_{odp(\overline{m})}\right)\right) \tag{5}$$

$$\lambda^{d}_{n+1} := \lambda^{d}_{n} + \rho^{d}_{n}\left(A^{d}\boldsymbol{D}(c^{n+1}_{od(\overline{m})}) - B^{d}\right) \tag{6}$$

$$\lambda^{s}_{n+1} := \lambda^{s}_{n} + \rho^{s}_{n}\left(C^{s}\boldsymbol{S}\left(x^{n+1}_{odp(\overline{m})}\right) - D^{s}\right) \tag{7}$$

$$\lambda^{l}_{n+1} := \lambda^{l}_{n} + \rho^{l}_{n}\left(c^{d}_{od(\overline{m})} - c^{s}_{od(\overline{m})}\right) \tag{8}$$

The iterations for minimizing and dualizing Eq. (3) is completed when primal feasibility and dual feasibility are achieved. The optimality conditions are formulated as

$$\text{Primal feasibility}: c^{d^*}_{od(\overline{m})} - c^{s}_{od(\overline{m})} = 0 \tag{9}$$

$$\text{Dual feasibility}: \nabla\boldsymbol{D}^{n+1}\left(c^{d^*}_{od(\overline{m})}\right) + \lambda^{d^*} = 0$$

$$\nabla\boldsymbol{S}^{n+1}(x^{s^*}_{odp(\overline{m})}) + \lambda^{s^*} = 0$$

To stop the iterative steps, the residuals of each feasibility with step size ρ were calculated. For instance, the corresponding residual for the primal feasibility function is expressed as:

$$\text{Primal residuals} r^{n+1} = c_{od(\overline{m})}^{d(n+1)} - c_{od(\overline{m})}^{s(n+1)} = 0 \tag{10}$$

where the primal and dual residuals converge to zero, and the defined feasibility conditions are also achieved when $n \rightarrow \infty$ (Yao et al., 2019).

3. Formulation of the integrated Demand-Supply optimization (IDSO) program

This section describes a sequence of reformulation steps for IDSO with the augmented Lagrangian relaxation function as the core solution algorithm in a stepwise manner.

M0: Original Form with Objective Functions and Constraints.

Without loss of generality, this study uses a logistic regression model to compute the number of destination choices and trips associated with alternative modes. For instance, the destination choices and number of trips selecting auto mode are expressed as follows:

$$TD_{od} = T_o \times \frac{e^{U_{(j)}}}{\sum_k e^{U_{(k)}}}, \forall o \tag{11}$$

$$f_{od(\overline{m})} = TD_{od} \times \frac{e^{-\theta c_{od(\overline{m})} + U_{(\overline{m})}}}{e^{-\theta c_{od(\overline{m})} + U_{(\overline{m})}} + \sum_{m \neq \overline{m}} e^{-\theta c_{od(m)} + U_{(m)}}}, \forall o, d \tag{12}$$

Eq. (12) represents the product of the total demand for origin o and the choice probability of a destination and derives the total demand for destination d, TD_{od}. $U_{(j)}$ is the utility function, which can be defined by the spatial separation between origin o and destination d and attractions at j (Fotheringham, 1986). In Appendix D, the convexity of the demand model has been described.

Depending on the generalized cost $c_{od(\overline{m})}$ and dispersion parameter θ, the potential number of trips/riders in selecting auto modes can be computed based on the generalized cost $c_{od(\overline{m})}$ and dispersion parameter θ. With the assumption that other behavioral parameters (i.e., socio-demographic characteristics) are given by choice modeling estimators (e.g., Bierlaire, 2003; Kim et al., 2021), the systematic utility function $U_{(\overline{m})}$, which measures the satisfaction in selecting the alternative mode, is assumed to be constant for simplicity. Eq. (12) denotes an essential form of the disutility/utility function, leading to the number of trips $f_{od(\overline{m})}$, from origin o to destination d with mode $\overline{m}$. To simplify the complexity of the decision-making processes, we first consider the integration of Eq. (11) using a network model.

To assign trip information in a network, the variational inequality (VI) condition is formulated to determine DUE states (Zhou et al., 2009; Lu et al., 2009). With path flow $x_{odp(\overline{m})}$ and path travel cost $t_{odp(\overline{m})}$, the optimal solutions that satisfy Wardrop's principle can be obtained as

DUE condition:

$$x_{odp(\overline{m})}\left(t_{odp(\overline{m})} - c_{od(\overline{m})} \ = \ 0, \forall o, d, p \in P(o,d)\right) \tag{13}$$

Minimum cost definitional constraint:

$$t_{odp(\overline{m})} - c_{od(\overline{m})} \geq 0, \forall o, \ p \in P(o,d)d, \tag{14}$$

where $c_{od(\overline{m})}$ is the minimum travel time cost from o to d with mode $\overline{m}$. The practical interpretation of Eq. (13) and (14) is that traffic path flows are assigned to least-cost paths until equilibrium states are reached across all possible paths (Lo and Chen, 2000). Moreover, to ensure path flow conservation and non-negativity for path flows, additional constraints are written as follows:

Flow conservation and non-negativity constraints:

$$\sum_{p \in P(o,d)} x_{odp(\overline{m})} = f_{od(\overline{m})}, \forall o, d \tag{15}$$

$$x_{odp(\overline{m})} \geq 0, \forall o, d, p \in P(o,d) \tag{16}$$

Eq. (15) indicates that the summation of path flows is equal to the number of trips obtained from Eq. (11). To clearly express the complex composite functions involved with the constraints, a fixed-point-based approach was employed. Next, the path travel time costs, $t_{odp(\overline{m})}$, are calculated, and an incidence matrix that connects the path layers and link layers is embedded to derive link performances and path travel time costs.

$$v_a = \sum_o \sum_d \sum_{p \in P(o,d)} x_{odp(\overline{m})} \times \delta_{odpa} \tag{17a}$$

$$\tau_a = FFTT_a \left(1 + \alpha \left(\frac{v_a}{\text{CAP}_a}\right)^{\beta}\right), \forall a \tag{17b}$$

$$t_{odp(\overline{m})} = \sum_{a} \delta_{odpa} \times \tau_a, \forall o, d, p \in P(o,d) \tag{17c}$$

$$c_{od(\overline{m})} = \min_{p \in P(o,d)} \{ t_{odp(\overline{m})} \} \tag{17d}$$

Eq. (17a)–(17d) illustrate the stepwise computation to obtain the final layer of the supply side variable as the path travel time cost. The first step computes the link volumes by using the results of Eq. (14), and the path-link incidence matrix δ_{odpa} (if link a is on path p, then 1; otherwise 0). The subsequent step uses the Bureau of Public Roads (BPR) function to obtain the link travel time τ_a, from origin node o to destination node d along path p. Finally, by multiplying the link-path incidence matrix δ_{odpa} and τ_a, we can derive the path travel time costs. This layered composite function then evaluates the minimum travel time cost $c_{od(\overline{m})}$ using $\min_{p \in P(o,d)} \{ \boldsymbol{t_{odp(\overline{m})}} \}$, which can be explained as a fixed-point formulation (Appendix A). Furthermore, to convert constrained programs with equality constraints to an unconstrained program, a reduced gradient method for a typical path flow-based formulation (Jayakrishnan et al., 1994; Chen et al., 2002) was used in this study. Instead of expressing the flow conservation in Eq. **(13)**, we consider the non-shortest path set as a non-basic variable and the shortest path flow as the basic variable to ensure that the number of variables can be reduced through the linear composition function $x_{od\overline{p}(\overline{m})} = f_{od(\overline{m})} - \sum_{p \in P(o,d); p \neq \overline{p}} x_{odp(\overline{m})}$. Thus, the constraints defined in Eq. (12)–(17d) in the original nonlinear optimization program are now expressed as a set of composite functions that are all embedded in the objective function.

Min: Eq. (13) written as

$$\sum_{o} \sum_{d} \sum_{p \in P(o,d)} x_{odp(\overline{m})} \left(t_{odp(\overline{m})} - c_{od(\overline{m})} \right) \tag{18a}$$

Subject to: variable-coupling equations Eq. (12)

$$= f_{od(\overline{m})} \tag{18b}$$

Eq. (15) written as

$$x_{od\overline{p}(\overline{m})} = f_{od(\overline{m})} - \sum_{\substack{p \in P(o,d) \\ p \neq \overline{p}}} x_{odp(\overline{m})}, \forall o, d \tag{18c}$$

Eq. (14).

Eq. (16).

Eq. (17a) – (17d).

In this optimization model, Eq. (18a) is the primary objective function with several variable coupling functions to link the demand and network models. Eq. (17a)–(17d) show that the link volume v_a is defined by the summation of the corresponding path flows $x_{odp(\overline{m})}$ with the path-link incidence matrix δ_{odpa}. The link travel time τ_a is then computed by the BPR function, and the path flow travel time $t_{odp(\overline{m})}$ is obtained by the product of τ_a and the link-path incidence matrix. Please note that to fulfill the shortest path-based assignment, we use the minimum function to find the lower cost in $t_{odp(\overline{m})}$, finding the lowest travel cost $c_{od(\overline{m})}$ (Jayakrishnan et al., 1994).

It is important to note that the proposed nonlinear model M0 can be extended to a rich set of demand and supply forms in different ways to code complex nonlinear functional forms. *First*, the relatively simple destination choice model (i.e., the travel demand component) can be extended to more realistic behavior models. For instance, the choice modeling structure can describe travelers' decision-making patterns in more detail, based on the activity-based framework studied by Bowman (1998). Using the concepts of downward (conditionality) and upward (accessibility), the activity-based modeling structure can be designed using a top-down modeling formulation. Specifically, by synthesizing different utility functions defined in the hierarchically layered system (long-term decision and daily scheduling decision, including tours and trip/stop), modelers can handle the conditionality of each choice model. Then, by defining log-sum variables (i.e., the expected maximum utility function), we can also design the accessibility between each layer. Such a nested system of discrete choice models can theoretically model travelers' activity sequences, where analytical behavioral models can be used.

Second, the volume delay function (VDF), expressed as the Bureau of Public Roads (BPR) function, can be formulated as a queue-evolution-based VDF (Belezamo, 2020) and other types of deterministic fluid-based models using polynomial time-dependent arrival rates (Newell, 1968; Newell, 1982). According to the proposed functional forms, the closed-form solutions from deterministic fluid-based queuing models can be systematically defined by computing the average system-wide delay as a function of queued demand and ultimate capacity, which can help us incorporate more realistic system performance models.

In summary, a detailed examination of such nonlinear functional forms found in travel demand modeling and network modeling can be applied to our approach and will be conducted.

M1: Variable splitting

In the **M0** module, the common variable appears in both systems, which means that $c_{od(\overline{m})}$ is indicated as the generalized cost in the choice model and the travel cost in the network model. The variable splitting method was used to handle the common variable simultaneously: $c_{od(\overline{m})} \rightarrow c^{d}_{od(\overline{m})}$ and $c^{s}_{od(\overline{m})}$.

Min: Eq. (18a) written as

$$\sum_{o}\sum_{d}\sum_{p\in P(o,d)} x_{odp(\overline{m})}\left(t_{odp(\overline{m})}-c^{s}_{od(\overline{m})}\right) \tag{19a}$$

subject to: variable-coupling equations Eq. (14) written as

$$t_{odp(\overline{m})}-c^{s}_{od(\overline{m})}\geq 0,\forall o,d,p\in P(o,d) \tag{19b}$$

Eq. (17d) written as

$$c^{s}_{od(\overline{m})}=\min\left\{t_{odp(\overline{m})}\right\} \tag{19c}$$

$$c^{d}_{od(\overline{m})}-c^{s=0:\text{ variable splitting constraint}}_{od(\overline{m})} \tag{19d}$$

Eq. (18b).
Eq. (18c).
Eq. (16).
Eq. (17a) – (17c).

M2: Dualize the constraints for the unconstrained optimization program

To determine the optimal solutions of the constrained optimization subject to equality and inequality constraints, we reformulate **M1** as an unconstrained optimization problem using the augmented Lagrange relaxation method. The three components are dualized: the VI condition, fixed-point formulations (demand–supply interaction functions), and variable splitting constraints. Eq. (19d) and Eq. (16) were dualized using Lagrangian multipliers and quadratic penalty functions. Furthermore, to compute composite variables, such as $c^{s}_{od(\overline{m})}$, we use the constraints in Eq. (17a)–(17c) and Eq. (19c).

$$\mathscr{L}_{AL}\left(x_{odp(\overline{m})},c^{d}_{od(\overline{m})};\boldsymbol{\lambda},\boldsymbol{\rho}\right)$$

$$=\sum_{o}\sum_{d}\sum_{p\in P(o,d)} x_{odp(\overline{m})}\left(t_{odp(\overline{m})}-c^{s}_{od(\overline{m})}\right)+\sum_{o}\sum_{d}\left[\lambda_{v}\left(c^{d}_{od(\overline{m})}-c^{s}_{od(\overline{m})}\right)+\frac{\rho_{v}}{2}\left(c^{d}_{od(\overline{m})}-c^{s}_{od(\overline{m})}\right)^{2}\right]+\sum_{o}\sum_{d}\sum_{p\in P(o,d)}\left[\lambda_{x}\left(x_{odp(\overline{m})}\right)+\frac{\rho_{x}}{2}\left(x_{odp(\overline{m})}\right)^{2}\right] \tag{20}$$

M3: Computational graph-based framework for the sequential architecture

Eq. (20) presents a variable-splitting-based augmented Lagrange relaxation framework with multipliers. To efficiently find optimal solutions in this layered structure with a large set of decision variables, we employ the principle of dynamic programming (DP) through the backpropagation mechanism. Eq. (20) is converted into a computational graph-based framework, translating the optimization problem into a sequentially layered architecture. The unconstrained program can be written as a system of nested composite functions (Recht, 2016):

$$\mathscr{L}_{CGAL}\left(x_{odp(\overline{m})},c^{d}_{od(\overline{m})};\boldsymbol{\lambda},\boldsymbol{\rho}\right)$$

$$=\sum_{o}\sum_{d}\sum_{p\in P(o,d)} x_{odp(\overline{m})}\left(t_{odp(\overline{m})}-c^{s}_{od(\overline{m})}\right)+\sum_{o}\sum_{d}\sum_{p\in P(o,d)}\left[\lambda_{v}\left(c^{d}_{od(\overline{m})}-c^{s}_{od(\overline{m})}\right)+\frac{\rho_{v}}{2}\left(c^{d}_{od(\overline{m})}-c^{s}_{od(\overline{m})}\right)^{2}\right]+\sum_{o}\sum_{d}\sum_{p\in P(o,d)}\left[\lambda_{x}\left(x_{odp(\overline{m})}\right)+\frac{\rho_{x}}{2}\left(x_{od\overline{m}}\right)^{2}\right] \tag{21}$$

$$\text{subject to} f_{od(\overline{m})}=g_{1}\left(c^{d}_{od(\overline{m})},\theta,TD_{o}\right) \tag{22a}$$

$$x_{odp'(\overline{m})}=g_{2}\left(f_{od(\overline{m})}\right),p'\in\{p,p'\neq\overline{p}\} \tag{22b}$$

$$x_{od\overline{p}(\overline{m})}=g_{3}\left(f_{od(\overline{m})},x_{odp'(\overline{m})}\right),\overline{p}\in\{p,\overline{p}\neq p'\} \tag{22c}$$

$$v_{a}=g_{4}\left(x_{odp'(\overline{m})},x_{od\overline{p}(\overline{m})},\delta_{odpa}\right) \tag{22d}$$

$$\tau_{a}=g_{5}(v_{a},FFTT_{a},\alpha,\beta,\text{CAP}_{a}) \tag{22e}$$

$$t_{odp(\overline{m})}=g_{6}\left(\tau_{a},\delta_{odpa}\right) \tag{22f}$$

$$c^{s}_{od(\overline{m})}=g_{7}\left(t_{odp(\overline{m})}\right) \tag{22 g}$$

where Eq. (22a)–(22g) indicate the variable-linking functions to compute the dualized optimization problem $\mathscr{L}_{CGAL}$, and these functions enable the powerful capability of constructing feedforward and backward computations through automatic differentiation (AD). Using this computational graph architecture, the gradients in Eq. (21), which help discover optimal solutions, can be calculated in a manner of dynamic programming (DP) and symbolical gradients. In other words, this critically important reformulation technique uses the backward propagation principle in widely available machine-learning packages, further reducing the effort required to compute gradients. The details of the computational graph-based functions expressed in Eq. (22a) – (22 g) are presented below.

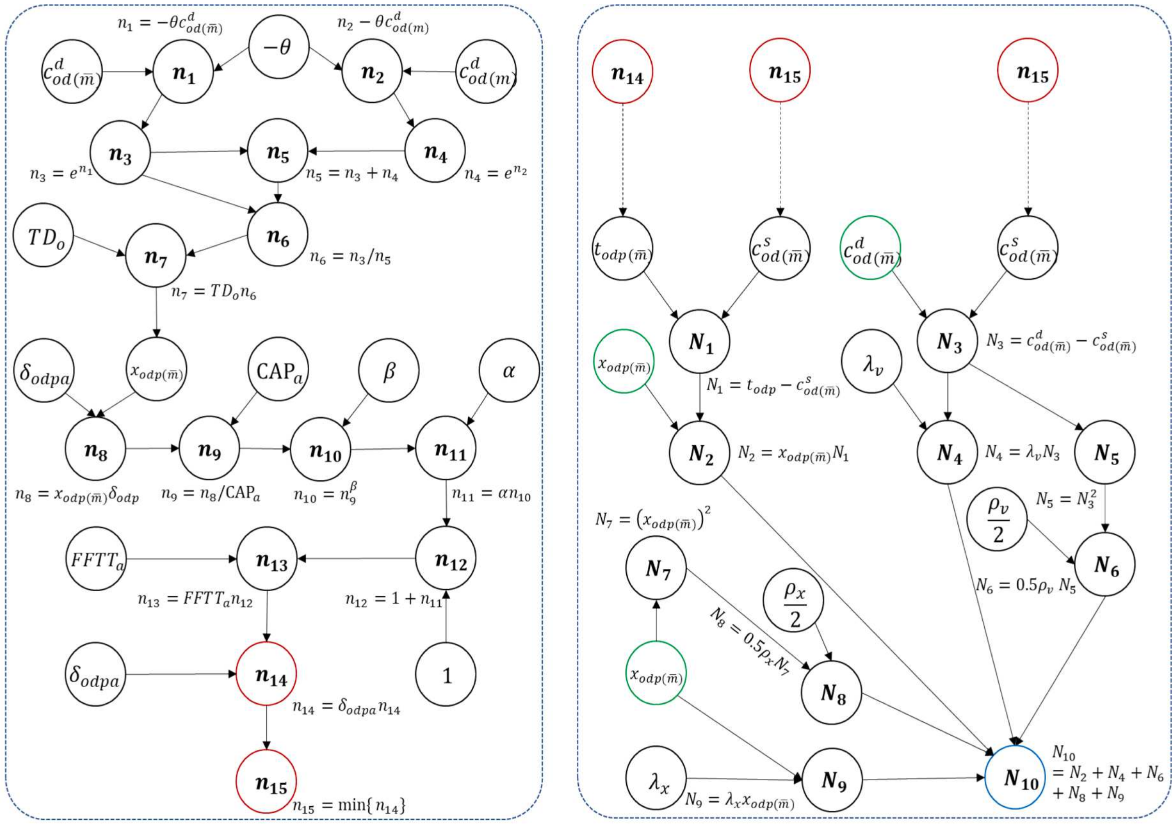
$g_1(\bullet)$	= Eq. (12)
$g_2(\bullet)$	= Fixed and non-shortest path flows, $x_{odp'\overline{m}} \geq 0$
$g_3(\bullet)$	= Fixed and shortest path flows, $x_{od\overline{p}\overline{m}} \geq 0$
$g_4(\bullet)$	= Eq. (17a)
$g_5(\bullet)$	= Eq. (17b)
$g_6(\bullet)$	= Eq. (17c)
$g_7(\bullet)$	= Eq. (19c)

Recent approaches for applying machine learning algorithms to efficiently compute higher-order gradients can be found in previous studies (Wu et al., 2016; Kim et al., 2021) for different problems, such as OD demand calibration and discrete choice model estimation.

As shown in Fig. 2, the sequential structure used to define the composite terminologies and unconstrained optimization problem, which presents the augmented Lagrangian expression in Eq. (21), can be described as follows. To simplify the expression for the graph-oriented structure, we considered one OD pair with two paths. The number of decision variables is one cost variable $c^d_{od(\overline{m})}$ and one path flow variable $x_{odp(\overline{m})}$ when applying the reduced-gradient algorithm, Eq. (18c). In Fig. 2, the $\boldsymbol{n}_7$ node defines the formulation in Eq. (12), we derived the number of trips for the OD pair. $\boldsymbol{n}_{14}$ and $\boldsymbol{n}_{15}$ nodes represent the path travel time cost $t_{odp(\overline{m})}$ and least travel time cost $c^s_{od(\overline{m})}$ and are substituted into Fig. 2 (b) as the input variables.

Remark 1. To approximate the perfect traffic assignment solution by modeling the path selection, we used the minimum functional form in the path travel time cost vectors $t_{odp(\overline{m})}$, selecting the lowest travel time cost value $c^s_{od(\overline{m})} = \min\{t_{odp(\overline{m})}\}$. Then, this computational graph-based augmented Lagrangian relaxation (CGLR) optimization function is expressed as:

$$\begin{aligned}\mathscr{L}_{CGAL} &= N_2 + N_4 + N_6 + N_8 + N_9 \\ &= x_{odp}N_1 + \lambda_v N_3 + \frac{\rho_v}{2}N_5 + \lambda_x x_{odp(\overline{m})} + \frac{\rho_x}{2}N_7 \\ &= x_{odp(\overline{m})}\left(t_{odp(\overline{m})} - c^s_{od(\overline{m})}\right) + \lambda_v\left(c^d_{od(\overline{m})} - c^s_{od(\overline{m})}\right) + \frac{\rho_v}{2}\left(c^d_{od(\overline{m})} - c^s_{od(\overline{m})}\right)^2 + \lambda_x x_{odp(\overline{m})} + \frac{\rho_x}{2}\left(x_{odp(\overline{m})}\right)^2\end{aligned} \tag{23}$$



(a) CG-based composite function for $t_{odp(\overline{m})}$ and $c^S_{od(\overline{m})}$

(b) CG-based augmented Lagrangian structure

Fig 2. Illustration of the Sequential Structure for the Composite Values and the Unconstrained Optimization Problem.

4. Solution algorithm

After translating the proposed optimization problem into a directed computational graph, the partial derivatives of Eq. (23) were calculated. Note that the simplified case is still used to intuitively explain the chain rules using gradients. With the assumption mentioned below, we implement AD based on the chain rule and intermediate nodes defined in the graph structure. As stated by Bartholomew-Biggs et al. (2000), AD operates the chain rule to generate analytic derivatives with respect to the given function, which guarantees the accuracy of the computed gradients. In addition, by building intermediate nodes in the original function (e.g., nodes in Fig. 2, such as N_2, N_4, or N_6 linking the input nodes and output nodes), it can decompose functions and improve computing performance for gradient-oriented optimization.

More importantly, to efficiently compute the iterative nature of this looping structure, we utilized the alternating direction method of multipliers (ADMM).

Assumption 1. The function, defined in Eq. (21) is a continuous function at any given point.

Assumption 2. The defined optimization problem has the local minimum.

With Eq. (21), and its computation graph is shown in Fig. 2. The partial derivatives of the augmented LR with respect to $x_{odp(\overline{m})}$ can be expressed as

$$\begin{aligned}\frac{\partial \mathscr{L}_{CGAL}}{\partial x_{odp(\overline{m})}} &= \frac{\partial N_{10}}{\partial N_9}\frac{\partial N_9}{\partial x_{odp(\overline{m})}} + \frac{\partial N_{10}}{\partial N_8}\frac{\partial N_8}{\partial N_7}\frac{\partial N_7}{\partial x_{odp(\overline{m})}} + \frac{\partial N_{10}}{\partial N_2}\frac{\partial N_2}{\partial x_{odp(\overline{m})}} \\ &= \lambda_x + \rho_x x_{odp(\overline{m})} + N_1 \\ &= \lambda_x + \rho_x x_{odp(\overline{m})} + t_{odp(\overline{m})} - c^s_{od(\overline{m})}\end{aligned} \tag{24}$$

Similarly, based on the chain rule method and the graph in Fig. 2, we can compute the gradients with respect to $c^d_{od(\overline{m})}$, and this computation is achieved using the backpropagation algorithm (reverse-mode automatic differentiation). Owing to this feedforward expression and backpropagation computation, the efficiency of computing a larger set of decision variables defined in the proposed model can be guaranteed (Olah, 2015).

To estimate different econometric choice models defined as non-concave logistic formulations (e.g., nested logit model), a quasi-Newton method (BFGS) was employed. Once the optimal solutions are found via the numerical optimizer, the primal problem and its constraints are computed to confirm the consistency/convergence criterion. If the method cannot reach satisfactory conditions, the Lagrangian multipliers are updated. Furthermore, we opted for a large starting value to avoid local minima or plateaus, thus initializing the multiplier close to 1. Using the exemplified equation, the updating process can be performed as follows:

$$\lambda_x^{k+1} = \max\left\{0, \lambda_x^k + \rho_x\left(x^k_{odp\overline{m}}\right)\right\} \tag{25}$$

$$\lambda_v^{k+1} = \max\left\{0, \lambda_v^k + \rho_v\left(c^d_{od(\overline{m})} - c^s_{od(\overline{m})}\right)\right\} \tag{26}$$

Eq. (25)–(26) denote the Lagrangian penalties involved with path flow positivity and demand–supply consistency, which can be renewed by the defined constraints and the quadratic penalty parameters, ρ_x and ρ_v. By updating these variables, the building block language-based program resolves the optimization problem by finding the minimum gap for the UE condition, demand–supply, and positivity for path flows. This iterative process is completed when the convergence criterion is reached, and the solving process for CGAL is illustrated below.

```
Algorithm: ADMM-based augmented Lagrangian method-based integration of demand and supply
Step 1: Initialization
    Load demand and network data
    Initialize iteration number $k = 0$;
    Initialize variables $c^{s(k)}_{od(\overline{m})}, t^{(k)}_{od(\overline{m})}, x^{(k)}_{odp(\overline{m})}$, and $c^{d(k)}_{od(\overline{m})}$
    Initialize Lagrangian multipliers $\lambda^k$ and set up quadratic penalty parameters $\rho$
Step 2: Minimize the augmented Lagrangian method over each component
    Step 2.1: Call the function $\mathscr{L}_{CGAL}(x_{odp(\overline{m})}, c^d_{od(\overline{m})}; \lambda, \rho)$ defined in Eq. (21)
    Step 2.2: Compute gradients of Eq. (21) using the AD algorithm
    Step 2.3: Send the results of Step 2.2 to the BFGS optimizer to solve decision variables,$x^{(k)}_{odp(\overline{m})}$and $c^{d(k)}_{od(\overline{m})}$
    For each component
        Find the optimal variable $x^{(k)}_{odp(\overline{m})}$
        Find the optimal variable $c^{d(k)}_{od(\overline{m})}$
        Compute the decision variables shown in Eq. (22a)–(22g)
    End for
Step 3: Compute the defined constraints
    Step 3.1: Compute the primal problem, Eq. (19a) with $x^{(k)}_{odp(\overline{m})}$
```

(*continued on next page*)

(*continued*)

```
Algorithm: ADMM-based augmented Lagrangian method-based integration of demand and supply
Step 1: Initialization
    Load demand and network data
    Initialize iteration number $k = 0$;
    Initialize variables $c^{s(k)}_{od(\overline{m})}, t^{(k)}_{od(\overline{m})}, x^{(k)}_{odp(\overline{m})}$, and $c^{d(k)}_{od(\overline{m})}$
    Initialize Lagrangian multipliers $\lambda^k$ and set up quadratic penalty parameters $\rho$
—
    Step 3.2: Compute the constraints, Eq. (19d) and Eq. (5) with the optimal variables $x^{(k)}_{odp(\overline{m})}$ and $c^{d(k)}_{od(\overline{m})}$
    Step 3.3: Compute the dualized optimization function, Eq. (10) with $x^{(k)}_{odp(\overline{m})}$ and $c^{d(k)}_{od(\overline{m})}$
Step 4: Evaluate the convergence criterion and constraints
    Set up the feasibility tolerance gap $\varepsilon_{gap}$ for BFGS (i.e., 1e-03)
    Check the relative gap $\frac{\mathscr{L}^{k+1}_{CGAL} - \mathscr{L}^{k}_{CGAL}}{\mathscr{L}^{k}_{CGAL}} \times 100\%$ with $\varepsilon_{gap}$
    Check the satisfaction of the defined conditions
    Update the Lagrangian multipliers, Eq. (25) – (26) if constraints are not satisfied*
```

Note*: If the solution does not satisfy the criterion defined in **Step 4, return to **Step 2**.

Remark 2. To have computational ease for the quadratic penalties, ρ, we use constant values (small numbers).

Remark 3. We set up the initial values of λ as zero.

Remark 4. To generate the number of feasible paths and travel time costs on those paths, we executed a traffic assignment package, DTALite, published by Zhou and Taylor (2014).

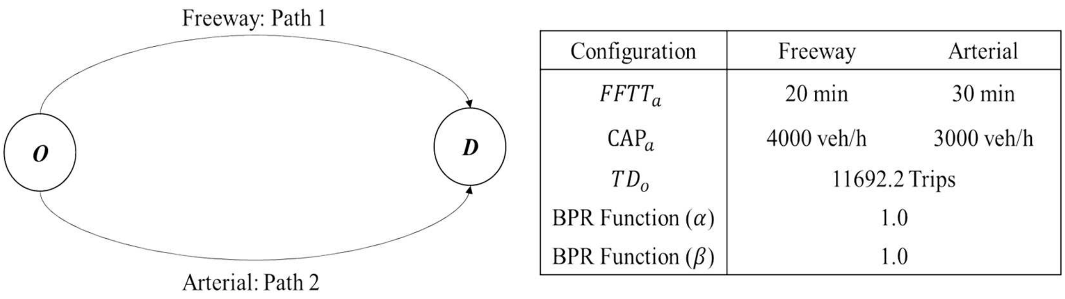
5. Numerical experiments

In this section, two sets of numerical experiments are conducted to demonstrate the feasibility of finding optimal solutions using the proposed computational graph-based optimization program and theoretically linking transportation demand and network models. The first experiment aimed to prove the accuracy of the solutions obtained using the gradient-based algorithm, while the second experiment evaluated the performance of a real road network (Chicago sketch network). Furthermore, we confirm that the algorithm can find network equilibrium states while fully linking travel demand and network assignment models. Please note that to create the initial values of path flows and path travel times, we utilize the column generation-based DTALite package published by Zhou and Taylor (2014) and assume that the total demand in each origin node is given.

5.1. *Small network (Iterative Solution, MSA, and gradient descent Algorithm)*

This subsection examines the computational accuracy of gradient-based optimization compared to an iterative method and the method of successive averages (MSA). Specifically, by comparing the reference values (path flows and travel costs) derived by an analytical formulation, we first checked the performance accuracy of our approach and the number of iterative steps to find optimal values. A simple corridor network includes two links (or paths) connecting one OD pair. The network attributes of a given corridor are shown in Fig. 3 and Eq. (12) is used to determine the travel mode choice behavior.

To compute the true reference values, an analytical formulation for identifying the user equilibrium (UE) conditions between two paths is derived. To readily derive the analytic expression, the parameter values of α and β in the BPR function were assigned as 1. Then, Eq. (17b) can be expressed as $\tau_a = FFTT_a(1 + (v_a/\text{CAP}_a))$, where v_a can be expressed as $x_{odp(\overline{m})}$, and the other path flow variable can be written as $f_{od(\overline{m})} - x_{odp(\overline{m})}$ using the reducing gradient algorithm. It should be remarked that, the above BPR function is typically used in static traffic assignment, while Zhou et al., (2022) recently proposed a *meso*-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to a BPR-based volume-delay functional form. Their macroscopic analytical formulations can be further incorporated to describe oversaturated system dynamics with consistent mesoscopic queue vehicular fluid model. By considering the UE condition, $\tau^F_a = \tau^A_a$, decision variables $x_{odp(\overline{m})}$ are calculated, and travel time costs $c_{od(\overline{m})}$ are also obtained.



Configuration	Freeway	Arterial
$FFTT_a$	20 min	30 min
CAP_a	4000 veh/h	3000 veh/h
TD_o	11692.2 Trips	
BPR Function (α)	1.0	
BPR Function (β)	1.0	

Fig 3. Small Network Illustrating the Configurations of Freeway and Arterial (Li et al., 2017).

$$FFTT_a^F\left(1+\frac{x_{odp(\overline{m})}}{\text{CAP}_a^F}\right)=FFTT_a^F\left(1+\frac{\left(f_{od(\overline{m})}-x_{odp(\overline{m})}\right)}{\text{CAP}_a^A}\right) \tag{27}$$

Solving Eq. (27), we can derive path flows $x_{odp(\overline{m})}$ to find UE, which is 5333.33 vehicles/hour, and travel time cost $c_{od(\overline{m})}$ is 46.67 min computed by the first order BPR function. Using these true reference values, we checked the convergence and performance accuracy of each benchmark method. To manage the common variable needed for both demand and supply, we split $c_{od(\overline{m})}$ into $c^d_{od(\overline{m})}$ and $c^s_{od(\overline{m})}$. Using the gradient descent method, we compute $c^d_{od(\overline{m})}$ and $c^s_{od(\overline{m})}$, while the benchmark follows the heuristic algorithm:

Iterative method:

$$c^d_{od(\overline{m})}(k+1)=c^s_{od(\overline{m})}(k) \tag{28}$$

MSA method:

$$c^d_{od(\overline{m})}(k+1)=\frac{k}{k+1}c^d_{od(\overline{m})}(k)+\frac{1}{k+1}c^s_{od(\overline{m})}(k) \tag{29}$$

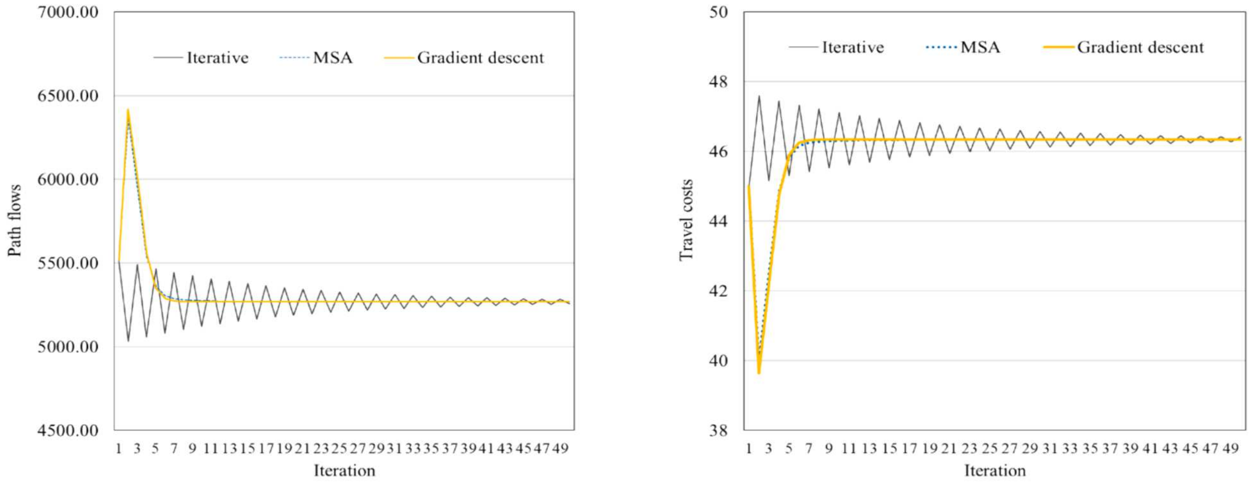
Analytical gradient descent method:

$$c^d_{od(\overline{m})}(k+1)=c^d_{od(\overline{m})}(k)-\alpha\nabla F\left(c^d_{od(\overline{m})}(k)\right) \tag{30}$$

where $\nabla F\left(c^{d(k)}_{od(\overline{m})}\right)$ is the gradient of the defined constraint function and $F\left(c^{d(k)}_{od(\overline{m})}\right)$ is defined as $\frac{1}{2}\left(c^{d(k)}_{od(\overline{m})}-c^{s(k)}_{od(\overline{m})}\right)^2$, enhancing the convexity of the given equation. Based on an arbitrary constant value (i.e., training parameter α) and the gradient:$\nabla F\left(c^{d(k)}_{od(\overline{m})}\right)=(c_d-c_s(c_d))\times\left(1-TD_o\times\left(\frac{FFTT_a^F}{3\text{CAP}_a^F}+\frac{FFTT_a^A}{2\text{CAP}_a^A}-\frac{FFTT_a^A}{3\text{CAP}_a^A}\right)\times\left(-\frac{\theta e^{\theta(c_d+c_t)}}{\left(e^{\theta c_d}+e^{\theta c_T}\right)^2}\right)\right)$, Eq. (30), was updated. With $c^d_{od(\overline{m})}(k)$, $c^s_{od(\overline{m})}(k)$, Eq. (12), and Eq. (17b), composite values were computed. In particular, $f_{od(\overline{m})}(k+1)$ was derived using Eq. (12) with $c^d_{od(\overline{m})}(k)$, $x_{odp(\overline{m})}(k+1)$ is given by Eq. (13) with $f_{od(\overline{m})}(k+1)$, andthe first-order BPR function with $x_{odp(\overline{m})}(k+1)$ calculates $\tau^F_a(k+1)$ and $\tau^A_a(k+1)$ values. Finally, by computing the average of their path travel times, we can determine the travel cost $c^s_{od(\overline{m})}(k+1)$.

Fig. 4 shows the convergence test between the three experiments. First, the iterative method shows the fluctuation in finding the solutions, and after 100 iterations, it derives the converged results, 5267.95 vehicles/hour and 46.34 min. The gradient descent method displays a shape similar to that of MSA in terms of finding optimal solutions. The computed variables are 5268.75 vehicles/hour and 46.34 min, almost identical to the results of MSA, demonstrating that the gradient-based methodology can find optimal solutions in different cases.

It should be noted that owing to the exemplified network and linearly defined UE condition, we can establish analytical formulations that cannot easily show the effectiveness of the gradient-based algorithm. If link performance functions are defined as nonlinear functions and the available paths increase, the network problem can be extremely complicated (Saitz, 1999). Although this experiment demonstrates that the major advantage of MSA is that it can avoid zigzagging displacements and does not respond to expensive line search operations such as those proposed by Zhou et al. (2009) and Ryu et al. (2017), there are critical limitations in finding optimal



(a) Computation of $x_{odp(\overline{m})}$

(b) Computation of $c^d_{od(\overline{m})}$

Fig 4. Convergence Comparison Test (Iterative Method, MSA, and Gradient Descent Algorithm.

path flow patterns and OD demand at a higher congestion level as well as a larger transportation network. Sbayti et al. (2007) stated that the MSA method, specifically applied in path-based formulations for medium-sized networks, results in explicit storage of path sets and path assignments, thereby increasing the number of computing iterations to find converged variables. Furthermore, in a more heterogeneous mixture of traffic conditions (e.g., time-variant dynamic congestions), MSA might be used as a valid starting point of deriving feasible equilibrium solution based on mesoscopic simulation or Macroscopic Fundamental Diagram (MFD) (Yildirimoglu and Geroliminis, 2014). On the other hand, because of the absence of a mathematical formulation for minimizing the relative gap in travel costs between inferior paths and current optimal (auxiliary) paths, MSA does not offer high-quality convergence under high congestion conditions.

Coupling MSA with a Block Gauss-Seidel decomposition, Florian et al. (2002) computed the network equilibrium formulation in one block and evaluated demand elements (trip distribution/mode choice) defined in the other block. This decomposition method can improve computing efficiency by reducing the number of iterative steps, but ADMM offers a well-defined analytical formulation of linking and decamping the two model components and holds the potential for solving complex transportation models with multiple integrated subsystems. It is worth noting that a partial linearization algorithm is another approach used to find optimal solutions for an integrated modeling system. Through direction-finding and line search, Oppenheim (1995) calculated the stochastic process and evaluated the objective function and its derivatives. As explained, this approach has shown the critical challenge and importance of handling the complexity of deriving the derivatives of logit-based functions. In other words, the CG-based algorithm can improve the efficiency of evaluating complex gradients analytically.

5.2. Comparing solutions based on mathematical programming solver and Graph-Based ADMM method

With the demonstration of the gradient-based approach, this experiment aims to assess the accuracy of solving the highly complex optimization problem, compared to standard optimization solvers using a mathematical programming language (AMPL) developed by Fourer et al. (1987). Programming the mathematical expression defined in Eq. (21) into AMPL and the graph-oriented ADMM approach, the objective function of finding the optimal path flows and travel costs is solved. In order to solve the nonconvex problem, AMPL uses the BARON global optimization solver, the graph-based ADMM method executes the iterative BFGS optimization. More detailed description of the global optimization solver BARON can be found in Sahinidis (1996). For this accuracy test, we sampled a set of the origin–destination pairs available in Chicago Sketch Transportation Network. Table 3 represents the optimization results of AMPL/BARON and the graph-based ADMM algorithm. In this example, we consider one origin node and multiple destination nodes; the number of the available origin–destination paths is nines. As shown in Table 3, both solvers find the optimal solutions of the path flows and travel costs while satisfying the user equilibrium condition.

We have recognized that there are decimal differences between the two programming tools, but in general, the optimal solutions from our approach are close to the global optimization results. This is important to note that the BARON solver enables us to efficiently solve this nonconvex problem, but the standard solver has encountered a difficulty in solving a large number of variables (i.e., path flows and travel costs) and obtaining the global solutions; the maximum number of the variables that can be solved by BARON for this complex integrated demand–supply structure was thirty-two variables. Through this experiment, we demonstrate the graph-oriented programming framework holds the potential to efficiently find optimal solutions for a reasonably large set of the demand/supply variables with complex nonlinear interactions.

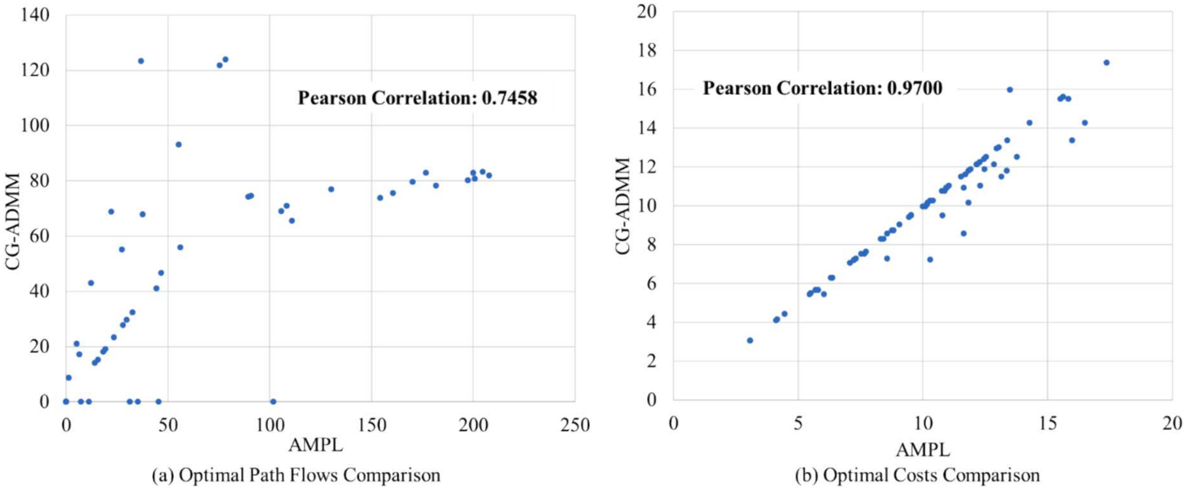
In this experiment, the solution of BARON was regarded as the global solutions and was used for the accuracy test of CG-ADMM with the Pearson correlation tests. As shown in Fig. 5, the path flows and costs from CG-ADMM were strongly correlated with the AMPL/BARON results, which were 0.7458 and 0.9700. More interestingly, the total CPU time of BARON used was 363.4 s on average, while CG-ADMM took only 41.66 s with a speed-up ratio of about 8 times.

5.3. Real-World case study (Chicago sketch transportation Network)

This subsection addresses the applicability of the computational graph-based optimization program in the Chicago transportation network, which is one of the highest transit ridership areas in the U.S. (Hughes-Cromwick and Dickens, 2018). Recognizing the importance of analyzing the impact of mode choice behaviors between personal vehicles and transit on the built network, we examine

Table 3
Optimization Results Comparison between AMPL/BARON and Graph-Oriented ADMM Approach.

Origin ID: 1		**AMPL/** BARON		**Graph-Oriented ADMM**		**Difference (%)**	
Destination	**Path**	**Path Flows,*x***	**Costs,*c***	**Path Flows,*x***	**Costs,*c***	**Total Path Flows**	**Total Costs**
1	p_1	181.61	3.06319	181.601	3.063023	0.097 %	−0.5375 %
2	p_2	207.82	4.10758	207.853	4.107368		
3	p_3	110.815	5.68181	110.828	5.681702		
	p_4	1.9167e-14	5.79119	−0.0002105	5.681725		
4	p_5	204.768	7.21798	204.806	7.218197		
5	p_6	108.274	5.49861	108.285	5.498685		
6	p_7	67.499	7.528	67.498	7.527913		
	p_8	6.2522e-14	7.63738	4.65769e-05	7.527915		
	p_9	3.1847e-14	7.70981	−9.3587e-06	7.637479		



(a) Optimal Path Flows Comparison

(b) Optimal Costs Comparison

Fig. 5. Optimal Path Flows/Costs Comparison between CG-ADMM and AMPL/BARON.

the capability of linking the behavior patterns and network models for the temporal equilibrium while finding the optimal solutions for the flow equilibrium. According to Fig. 6, the number of nodes is 933, and the number of links is 2,950. In addition, under a congested traffic state, the number of generated paths is 285,959, and the number of origin–destination pairs is 142,890. Therefore, the total number of path flow variables $x_{odp(\overline{m})}$ and travel costs $c^{d}_{od(\overline{m})}$ to link the demand and network models is 428,849, and the size of the path-link incidence matrix is 285,959 × 2,950. In this study, to evaluate the feasibility of the proposed mathematical optimization program and avoid the computational burden of finding a large set of decision variables, we selected a few pairs of origin–destination matrices pairs. Consequently, the number of $x_{odp(\overline{m})}$ to be computed is 3,835, and the number of $c^{d}_{od(\overline{m})}$ is 1,012, as shown in Fig. 6, where the colored blue lines indicate path flow sequences. The dataset of this network can be downloaded from the following website: https://github.com/asu-trans-ai-lab/Path4GMNS/tree/master/dataset.

In this study, five different origin nodes are selected. We sequentially find optimal solutions for each origin–destination pair using the CG-ADMM method. To validate this optimization algorithm, we tested the convergence of the objective function through Eq. (21) while achieving the satisfaction of the constraints. In other words, the three components of minimizing the optimization problem are tested: the “bridge gap”, which assures the linkage constraint of demand–supply units in Eq. (19d), the user equilibrium gap, which presents network equilibrium states; and the path flow positivity in Eq. (16).

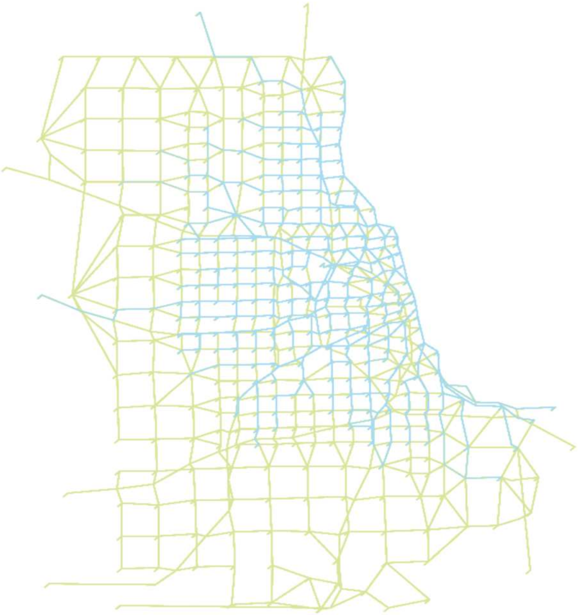


Fig 6. Chicago Sketch Transportation Network.

In Fig. 7, each subplot indicates the results of the defined constraints and the convergence of the objective function. We first illustrate the result from origin id 1 with multiple destination ids; the number of *od* pairs is 229 and the paths are 639. Fig. 7(a) shows the relative gap between demand and supply. If the optimized variables (i.e., $c^d_{od(\overline{m})}$ and $c^s_{od(\overline{m})}$) are not close to each other, the variance of the two variables is largely distributed. In this case, except for a few pairs, the proposed model was able to find consistent solutions. Fig. 7(b) represents the user equilibrium condition, where most of the measured data points are found near zero, displaying the equilibrium state. The average UE gap in this scenario was 0.058. Furthermore, Fig. 7(c) shows the positivity of the path flows $x_{odp(\overline{m})}$. The convergence of the objective function is shown in Fig. 7(d). The number of iterations of the BFGS algorithm was 500, and we set the tolerance value to 0.001. This condition indicates that if the relative gap between the previous step and the current step is smaller than the tolerance, then the numerical optimizer is stopped. Checking for the satisfaction of the restraints, we decide whether to update the Lagrangian multipliers.

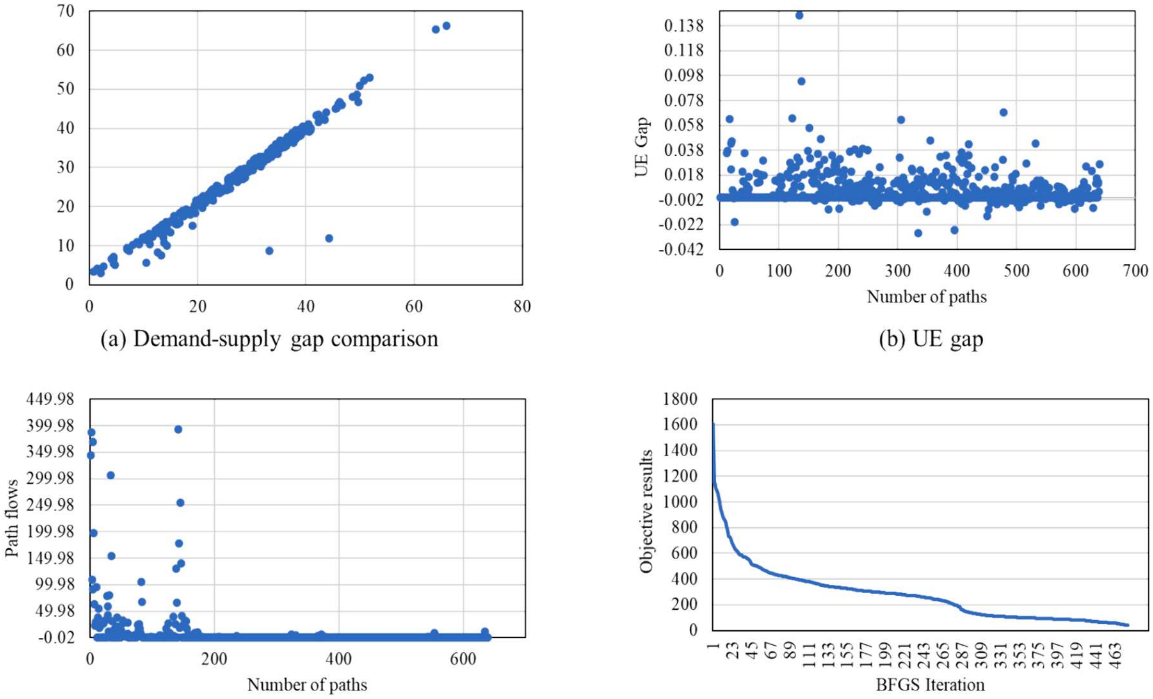
Assuming the mutual exclusivity of each origin–destination pair, we execute numerical experiments on different OD pairs. The configuration setting for the BFGS algorithm followed the aforementioned values, and the average computing time for solving the objective function and obtaining optimal solutions was approximately 80 s. The average values of each constraint, namely, the UE gap, “cost-bridge” gap, and path flow positivity, are listed in Table 4.

5.4. Real-World case study (Chicago sketch transportation Network) with Beckmann’s formulation

The last subsection of the chapter introduces how a large set of unknown variables can be solved through a newly suggested formulation. From the adopted objective function (gap-based formulation), we have recognized the limitation of finding optimal path flow variables. For instance, the Chicago sketch network has 285,959 path flows between origin-to-destination pairs, which could be a hard optimization problem such as calculation of the inversed hessian matrix with the large set of the variables while satisfying the dualized constraints in Eq. (20). To handle the difficulty, Beckmann’s formulation has been adopted in this numerical experiment, which can guarantee the convexity of the link performance function and enable us to reduce the number of the constraints defined in the objective function (Beckmann et al., 1956). Instead of solving the previously defined objective function (Eq. (20)), the user equilibrium solution for the path flow-based objective function can be expressed as: Table 5.

$$\mathscr{L}_{AL}\left(x_{odp(\overline{m})};\lambda,\rho\right) = \sum_{(i,j)\in A}\int_0^{\sum_{p\in P(o,d)}\delta^p_{ij}x_{odp(\overline{m})}} t_{ij}(w)dw + \sum_o\sum_d\sum_{p\in P(o,d)}\left[\lambda_x\left(x_{odp(\overline{m})}\right)+\frac{\rho_x}{2}\left(x_{odp(\overline{m})}\right)^2\right] \tag{31}$$

Eq. (31) presents the reformulated Beckmann’s formulation with the path flow positivity constraint, finding the optimal path flow variables through the link performance expression defined as the BPR function. δ^p_{ij} is the incidence matrix to convert path flows into



(a) Demand-supply gap comparison

(b) UE gap

(c) Path flow positivity

(d) Objective results

Fig 7. CG-ADMM Results of Origin Node 1.

Table 4
Optimization Results: Five Different Origin Nodes.

IDs	OD Pairs (Paths)	Time	User Equilibrium Gap			Path Flow Positivity			Demand-Supply Consistence gap		
			Min	Max	Avg	Min	Max	Avg	Min	Max	Avg
1	229 (639)	75 s	−0.04	11.02	0.06	−0.02	392.71	7.53	−2.79	32.49	−0.00
2	236 (763)	81 s	−0.12	48.48	0.17	−0.03	731.93	8.66	−4.20	10.18	0.00
3	247 (730)	80 s	−0.94	8.30	0.04	−0.16	1422.13	13.07	−3.71	55.18	0.00
4	250 (902)	83 s	−1.03	9.68	0.06	−0.17	990.58	9.94	−10.9	61.75	0.00
5	279 (801)	77 s	−0.48	22.98	0.08	−0.08	3033.68	20.44	−13.6	36.0	−0.00

Table 5
Formulation for Interactive Variables (i.e., Costs on Demand Side and Supply Side).

Solution algorithm	Formulation	Step size	Publication
Method of successive average	$c^d_{k+1} = \frac{k-1}{k} c^d_k + \frac{1}{k} c^s_k$	Iterative steps $(1/k)$	Florian et al. (2002), Lin et al. (2008), Zhou (2008)
Constant weight	$c^d_{k+1} = (1-w)c^d_k + wc^s_k$	Fixed (w)	Boyce et al. (2008)
Gradient projection	$c^d_k = c^s_k(x_k) + \frac{1}{\beta_r}\ln(x_k)$	Self-adaptive	Zhou et al. (2009)
Descent direction	$c^d_k = t_k\left(x\left(f_k(\bullet)\right), \delta\right)$	Line search	Ryu et al. (2017)
Analytical gradients using AD	$c^d_k = c^s_k$	Lagrangian penalty	This paper

link flows. It should be noted that there are two critical challenges, multiplication of the large set of the unknown variables and computation of the inversed matrix. As presented, the number of unknown variables is close to 300,000 variables which cause the out of memory issue (e.g., memory usage can be 300,000 × 300,000 × 32bits = 2.88 $\times 10^{12}$ bits or 360 gigabytes) and increase the complexity of computing matrices. In light of this, we have re-structured the incidence matrix as an array-based form (e.g., 300,000 × 1) to minimize the memory usage and reshaped the matrix as a spare-dense matrix that can significantly reduce the memory and complexity of the problem. In addition, the limited-memory BFGS has been used to avoid saving a dense inversed Hessian matrix (n × n).

Refactoring the formulations and implementing the new optimization algorithm, we have found the optimal path flows that can more minimize the relative gap compared to DTALite with a highly efficient computing time. Fig. 8 presents the improved results of the relative gap function, which can potentially help us find more accurate optimal solutions when compared to the DTALite solutions. As shown, the initial relative gap was the same as the DTALite result, but ADMM-based Beckman objective function with the analytically computed gradients (*automatic differentiation*) enables us to find the smaller relative gap within 10 iterations. That is, the graph-

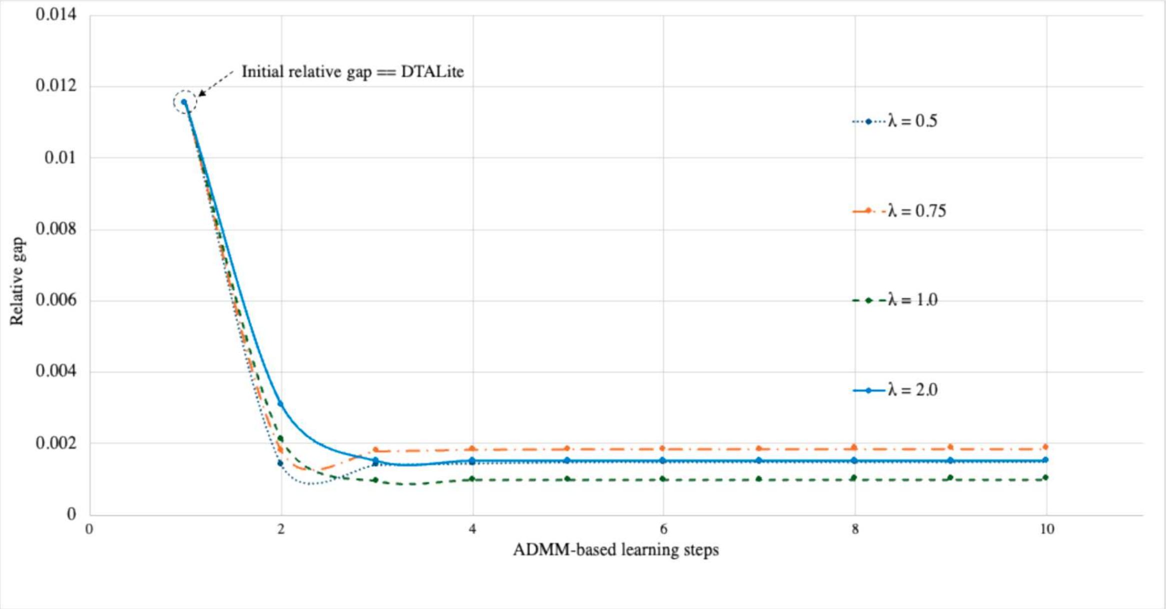


Fig 8. Relative Gap Comparison between CG-Based Optimization and DTALite.

oriented reformulation for objective functions can be extended to link different modeling components and possibly solve more complex structural problems (e.g., analytical solutions to find and integrate demand–supply equilibrium). In addition, by initializing different ADMM multipliers, we evaluated the robustness of the proposed formulation. As shown in Fig. 8, each multiplier followed similar convergence patterns, and the multiplier λ with a value of 1.0 showed the lowest relative gap results compared to the other multipliers.

6. Summary and conclusions

In this study, by developing analytical/gradient tractable structures with computational graphs, we propose an augmented Lagrangian relaxation framework (CG-ADMM) to find primal and dual solutions for a consistent integration of demand–supply models. First, we construct a variational inequality formulation and constraints to follow Wardrop's principle. In this mathematically-oriented reformulation framework, by splitting the common variable used in both the demand and supply sides, we created linking constraints that can minimize the gap between demand and supply components. To compute the gradients of the decision variables (i.e., optimal path flows and traveling costs) of the objective function, automatic differentiation (AD), specifically the reverse-mode AD algorithm, was conducted. Transmitting the gradients into the numerical optimizer (BFGS), we found the optimal decision variables, with the program constantly updating the Lagrangian multipliers to minimize the UE gap and bridge gap (demand–supply consistency gap) while maintaining path flow positivity. To demonstrate the effectiveness of the proposed approach, numerical experiments were conducted, including a small network and the Chicago sketch transportation network. The computational graph-based augmented Lagrangian relaxation (CG-ADMM) method can provide effective computational methods for solving this class of highly nonlinear problems involving complex demand–supply interactions.

Compared to prior studies in Table 4, our proposed method, including the consistent coupling constraint, would be a promising model reformulation framework that providing sufficient descriptive ability for interactive transportation systems.

The coupling formulations presented in previous literature in Table 1 are not modeled explicitly as constraints (e.g., gradient projection or descent direction). Instead, they are indirectly enforced through optimality conditions in the final stage, resulting in the insufficient establishment of highly interactive transportation systems. In comparison, our proposed mathematical programming model specifically includes and further dualizes the coupling condition to better measure demand and supply interactions during the solution-finding process.

Overall, although transportation modelers have developed comprehensive modeling structures with heuristic algorithms and follow iterative sequential processes to solve mathematical objective functions, find optimal path flows, and travel costs to satisfy equilibrium states, our proposed methodology aims to advance the kernel of coupling/developing interactive transportation systems by providing (1) a consistent mechanism to mathematically integrate the accessibility cost on the demand side and the travel cost computed by traffic assignment; (2) gradient-oriented approximation to solve complex objective functions composed of the costs, path flows, OD flow patterns, and choice probabilities, and (3) the computational power of calibrating a large set of decision variables in a real transportation network. In other words, determining step sizes based on the analytical gradients could offer a numerically stable solution-finding approach compared to simple "averaging" manner. Furthermore, the included analytically derived dynamic supply-side functions could better mitigate the inconsistency between the demand and supply variables in terms of flow, cost, and choice probability. Finally, solution algorithms designed to directly minimize and reduce the model gap measure iteratively could be extended to solve more complex integrated travel demand and network modeling architectures.

With this graphically oriented mathematical modeling framework and its solution algorithm using automatic differentiation (AD), our research further focuses on two aspects:

- Extension of the modal split-traffic assignment modeling structure to consider more realistic scenarios (behavioral richness and spatio-temporal dimension): As stated in studies by Zhou et al. (2009) and Ryu et al. (2017), to have more behavioral richness in a top-down sequential system, a hierarchical choice structure will be defined by satisfaction functions or log-sum terms and constructed using computational graphs to calculate the analytical gradients with respect to decision variables. By doing so, we can generate more realistic behavioral coupling constraints in the model while ensuring computational efficiency in dealing with highly complex composite terminologies. Furthermore, for consistency in the representation of behavioral units, spatial relationships, and temporal scales (Pendyala et al., 2012), spatial–temporal constraints must be dualized in a potential modeling framework (Liao et al., 2013; Chow, 2018; Mahmoudi et al., 2021, Xiong et al., 2021).
- Problem decomposition for a larger-scale network experiment: With high-performance computing techniques used for the calibration of a large set of decision variables, we will evaluate the applicability of a programming architecture at the national level, capturing a wide range of emerging patterns in the transportation ecosystem (Auld et al., 2016). We would study different types of quasi-Newton methods, such as limited-BFGS or a first-order method and the ADAM optimization algorithm (Kingma and Ba, 2014), to improve the computing efficiency of finding optimal flow patterns, costs, and decision makers' choice probability while satisfying consistency in the representation listed above. Consequently, we can further address a broader set of transportation system modeling challenges, land use-transport interaction (Acheampong and Silva, 2015), congestion, emergency response planning, and toll/incentive issues in built network environments.

7. Model implementation

This proposed framework has been implemented based on the Python code, which can be downloaded at https://github.com/Taehooie/CGEquilibrium. In this folder, users can find the transportation network datasets as well.

CRediT authorship contribution statement

Taehooie Kim: Conceptualization, Data curation, Formal analysis, Methodology, Project administration, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jiawei Lu:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Writing – review & editing. **Ram M. Pendyala:** Conceptualization, Funding acquisition, Investigation, Supervision, Writing – original draft, Writing – review & editing, Project administration. **Xuesong Simon Zhou:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Validation, Writing – original draft, Writing – review & editing, Visualization.

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.

Acknowledgements

Partial support for this research was provided by the Center for Teaching Old Models New Tricks (TOMNET), which is a Tier 1 University Transportation Center sponsored by the US Department of Transportation under grant 69A3551747116. The second and last authors were partially supported by the National Science Foundation, United States, under Grant No. CMMI 1663657, "Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks".

Appendix A. . Fixed-Point formulation of the interaction between transportation demand and supply

$$x = \text{feasible path flows obtained traffic assignment package (Zhou and Taylor, 2014)} \tag{A.1}$$

$$\tau(x,\alpha,\beta) = \text{link traveling cost} \tag{A.2}$$

$$t(\tau(x,\alpha,\beta)) = \text{path traveling cost} \tag{A.3}$$

$$\mathrm{MNL}(t(\tau(x,\alpha,\beta));\theta,U) = \text{choice behavior resulting from (A.3)} \tag{A.4}$$

$$f(\mathrm{MNL}(t(\tau(x,\alpha,\beta));\theta,U),\mathrm{OD}) = \text{path flows based on OD demand and (A.4)} \tag{A.5}$$

$$\delta(f(\mathrm{MNL}(t(\tau(x,\alpha,\beta));\theta,U),\mathrm{OD})) = \text{path cost mapped by the incidence matrix} \tag{A.6}$$

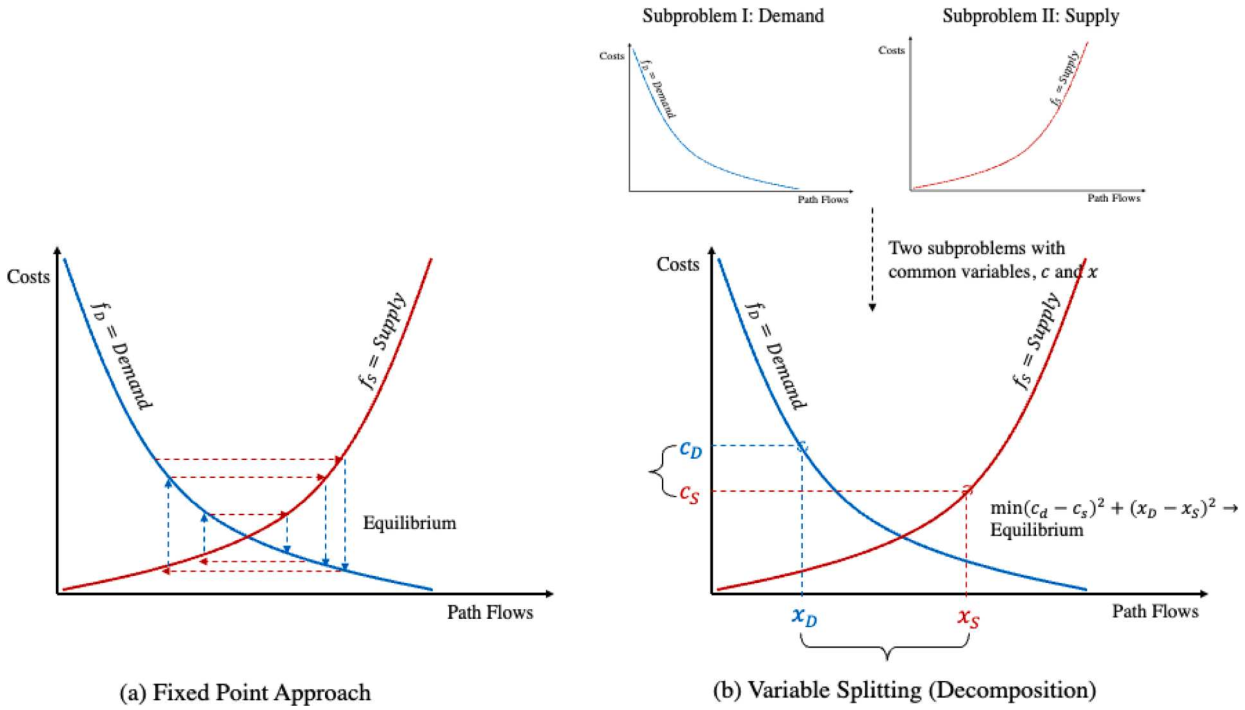
$$t(\tau(x,\alpha,\beta)) = \delta(f(\mathrm{MNL}(t(\tau(x,\alpha,\beta));\theta,U),\mathrm{OD}))\text{the fixed} - \text{point formulation }(x \geq 0 \text{ and } x \in D) \tag{A.7}$$

Appendix B. . Difference between automatic differentiation and the chain rule in computing gradients of a composite function

The fundamental process of decomposing differentials is similar to the chain rule approach; however, there are two different features that distinguish AD from the chain rule. As stated in Bartholomew et al. (2000), AD carries floating-point numerical values instead of differentiating the symbolic expressions decomposed and reduces the complexity of computing the complex composite function. Furthermore, by using intermediate variables as checkpoints, the AD algorithm can save memory. For instance, we provide a simple example, $h(x) = f(g(k(x)))$ and show how intermediate variables are defined. By defining intermediate variables w_i in each decomposed element, we can store the computed gradients and carry numerical values. However, as the pure chain rule approach needs to compute the symbolic expressions without the stored variables (i.e., intermediate variables), the computational cost is much higher than that of AD.

Automatic Differentiation (AD)	*Chain Rule*
$\frac{\partial h(x)}{\partial x} = \frac{\partial f(w_2)}{\partial w_2}\frac{\partial w_2}{\partial w_1}\frac{\partial w_1}{\partial w_0}$ where,$w_0 = x$ $w_1 = k(w_0)$ $w_2 = g(w_1)$ $w_3 = f(w_2) = h(x)$	$\frac{\partial h(x)}{\partial x} = f'(g(k(x)))g'(k(x))k(x)$

Appendix C. . Illustration: Fixed point approach and variable splitting (Decomposition)



(a) Fixed Point Approach

(b) Variable Splitting (Decomposition)

Fig. C1. Optimization Algorithms: Fixed Point and Variable Splitting

Appendix D. . Convexity of demand model

$$f_{od(\overline{m})} = TD_{od} \times \frac{e^{-\theta c_{od(\overline{m})}+U_{(\overline{m})}}}{e^{-\theta c_{od(\overline{m})}+U_{(\overline{m})}} + \sum_{m\neq\overline{m}} e^{-\theta c_{od(m)}+U_{(m)}}}, \forall o, d \tag{C.1}$$

The convexity of the demand model can be proven by showing the positivity of the second derivative of Eq (12), the logistic probability formulation. For the simplification of the differentiation of the equation, assuming $\theta \rightarrow 1$ and omitting the utility expression of $U_{(\overline{m})}$, the given equation was reformulated as:

$$P_{od(\overline{m})} = \frac{e^{-c_{od(\overline{m})}}}{e^{-c_{od(\overline{m})}} + e^{-c_{od(m)}}} \tag{C.2}$$

Differentiating the function with respect to $c_{od(\overline{m})}$, we can compute the first order derivative of C.2; in this logistic, we consider the two-mode choice behavior (auto versus non-auto)

$$\frac{\partial P_{od(\overline{m})}}{\partial c_{od(\overline{m})}} = -\frac{\left(e^{c_{od(m)}+c_{od(\overline{m})}}\right)}{\left(e^{c_{od(\overline{m})}} + e^{c_{od(m)}}\right)^2}, \{c_{od(\overline{m})} \in \mathbb{R} : e^{c_{od(m)}+c_{od(\overline{m})}} \neq 0\} \tag{C.3}$$

Because of the continuity and adaptation of automatic differentiation, the second derivative of C.3 can be derived by the multiplication of the above expression:

$$\begin{aligned} \frac{\partial}{\partial c_{od(\overline{m})}} &= \left(-\frac{\left(e^{c_{od(m)}+c_{od(\overline{m})}}\right)}{\left(e^{c_{od(\overline{m})}} + e^{c_{od(m)}}\right)^2}\right) \\ &= \frac{\left(e^{c_{od(m)}+c_{od(\overline{m})}}\right)\left(e^{c_{od(m)}} - e^{c_{od(\overline{m})}}\right)}{\left(e^{c_{od(\overline{m})}} + e^{c_{od(m)}}\right)^3} \end{aligned} \tag{C.4}$$

The positivity of the second order derivative can be true if $c_{od(m)}$ exists in the positive domain.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. and Kudlur, M., 2016. Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), pp. 265–283.

Acheampong, R.A., Silva, E.A., 2015. Land use–transport interaction modeling: a review of the literature and future research directions. J. Transp. Land Use 8 (3), 11–38.

Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., Zhang, K., 2016. POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. Transp. Res. Part C: Emerg. Technol. 64, 101–116.

Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2017. Automatic differentiation in machine learning: a survey. J. Mach. Learn. Res. 18 (1), 5595–5637.

Beckmann, M., McGuire, C.B., Winsten, C.B., 1956. Studies in the. Econ. Transp. No, 226 pp).

Bierlaire, M., 2003. BIOGEME: A free package for the estimation of discrete choice models. In Swiss Transport Research Conference (no. CONF).

Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., Oliveira, F., 2018. Combining Progressive Hedging with a Frank Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed Integer Programming. SIAM J. Optim. 28 (2), 1312 1336.

Boyce, D., Ralevic-Dekic, B., Bar-Gera, H., 2004. Convergence of traffic assignments: how much is enough? J. Transp. Eng. 130 (1), 49–55.

Boyce, D., O'Neill, C.R., Scherr, W., 2008. Solving the sequential travel forecasting procedure with feedback. Transp. Res. Rec. 2077 (1), 129–135.

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning 3 (1), 1–122.

Boyles, S. D., Lownes, N.E., Unnikrishnan, A., 2022. Transportation Network Analysis, Volume I, Version 0.90.

Cantarella, G.E., Cartenì, A., de Luca, S., 2015. Stochastic equilibrium assignment with variable demand: theoretical and implementation issues. Eur. J. Oper. Res. 241 (2), 330–347.

Cantelmo, G., Viti, F., 2019. Incorporating activity duration and scheduling utility into equilibrium-based Dynamic Traffic Assignment. Transp. Res. B Methodol. 126, 365–390.

Chen, A., Lee, D.H., Jayakrishnan, R., 2002. Computational study of state-of-the-art path- based traffic assignment algorithms. Math. Comput. Simul 59 (6), 509–518.

Chow, J., 2018. Informed Urban Transport Systems: Classic and Emerging Mobility Methods Toward Smart Cities. Elsevier.

Chu, Y.L., 2018. Implementation of a new network equilibrium model of travel choices. J. Traffic Transp. Eng. (Engl. Ed.) 5 (2), 105–115.

Eckstein, J., Bertsekas, D.P., 1992. On the Douglas Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. 55 (1 3), 293 318.

Esser, J., Nagel, K., 2001. Iterative demand generation for transportation simulations. Lead. Edge Travel Behav. Res. 659–681.

Fisher, M.L., Jörnsten, K.O., Madsen, O.B., 1997. Vehicle routing with time windows: two optimization algorithms. Oper. Res. 45 (3), 488–492.

Florian, M., Wu, J.H., He, S., 2002. A multi-class multi-mode variable demand network equilibrium model with hierarchical logit structures. In: Transportation and Network Analysis: Current Trends. Springer, Boston, MA, pp. 119–133.

Flötteröd, G., Chen, Y., Nagel, K., 2012. Behavioral calibration and analysis of a large-scale travel microsimulation. Netw. Spat. Econ. 12 (4), 481–502.

Fourer, R., Gay, D.M., Kernighan, B.W., 1987. AMPL: A mathematical programming language. AT & T Bell Laboratories, Murray Hill, NJ.

Guarda, P., Battifarano, M., Qian, S., 2024. Estimating network flow and travel behavior using day-to-day system-level data: a computational graph approach. Transp. Res. Part C: Emerg. Technol. 158, 104409.

Guignard, M., 2003. Lagrangean relaxation. Top 11 (2), 151–200.

Habib, K.N., Sasic, A., Weis, C., Axhausen, K., 2013. Investigating the nonlinear relationship between transportation system performance and daily activity–travel scheduling behaviour. Transp. Res. A Policy Pract. 49, 342–357.

Halat, H., Mahmassani, H.S., Zockaie, A., Vovsha, P., 2017. Activity cancellation and rescheduling by stressed households: improving convergence in integrated activity-based and dynamic traffic assignment models. Transp. Res. Rec. 2664 (1), 100–109.

Han, Y., Pereira, F.C., Ben-Akiva, M., Zegras, C., 2022. A neural-embedded discrete choice model: learning taste representation with strengthened interpretability. Transp. Res. B Methodol. 163, 166–186.

Hao, J.Y., Hatzopoulou, M., Miller, E.J., 2010. Integrating an activity-based travel demand model with dynamic traffic assignment and emission models: Implementation in the Greater Toronto, Canada, area. Transp. Res. Rec. 2176 (1), 1–13.

Hughes-Cromwick, M., Dickens, M., 2018. APTA 2017 Public Transportation Fact Book.

Jayakrishnan, R., Tsai, W.T., Prashker, J.N., Rajadhyaksha, S., 1994. A faster path-based algorithm for traffic assignment.

Kang, J.E., 2013. Integration of Locational Decisions with the Household Activity Pattern Problem and Its Applications in Transportation Sustainability. University of California, Irvine. Approach for assessing the impacts of advanced traffic information under realistic stochastic capacity distributions. Transp. Res. Part C: Emerg. Technol. 77, 113–133.

Kim, T., Sharda, S., Zhou, X., Pendyala, R.M., 2020. A stepwise interpretable machine learning framework using linear regression (LR) and long short-term memory (LSTM): City-wide demand-side prediction of yellow taxi and for-hire vehicle (FHV) service. Transp. Res. Part C: Emerg. Technol. 120, 102786.

Kim, T., Zhou, X., Pendyala, R.M., 2021. Computational graph-based framework for integrating econometric models and machine learning algorithms in emerging data-driven analytical environments. Transportmetr. A: Transp. Sci. 1–35.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Konduri, K.C., 2012. Integrated Model of the Urban Continuum with Dynamic Time-dependent Activity-travel Microsimulation: Framework, Prototype, and Implementation. Arizona State University.

Lam, W.H., Huang, H.J., 2003. Combined activity/travel choice models: time-dependent and dynamic versions. Netw. Spat. Econ. 3 (3), 323–347.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Li, M., Rouphail, N.M., Mahmoudi, M., Liu, J., Zhou, X., 2017. Multi-scenario optimization.

Liao, F., Arentze, T., Timmermans, H., 2013. Incorporating space–time constraints and activity-travel time profiles in a multi-state supernetwork approach to individual activity-travel scheduling. Transp. Res. B Methodol. 55, 41–58.

Lin, D.Y., Eluru, N., Waller, S.T., Bhat, C.R., 2008. Integration of activity-based modeling and dynamic traffic assignment. Transp. Res. Rec. 2076 (1), 52–61.

Liu, J., Kang, J.E., Zhou, X., Pendyala, R., 2018. Network-oriented household activity pattern problem for system optimization. Transp. Res. Part C: Emerg. Technol. 94, 250–269.

Liu, Y., Tong, L.C., Zhu, X., Du, W., 2021. Dynamic activity chain pattern estimation under mobility demand changes during COVID-19. Transp. Res. Part C: Emerg. Technol. 131, 103361.

Lo, H.K., Chen, A., 2000. Reformulating the traffic equilibrium problem via a smooth gap function. Math. Comput. Model. 31 (2–3), 179–195.

Lu, C.C., Mahmassani, H.S., Zhou, X., 2009. Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. Transp. Res. B Methodol. 43 (3), 345–364.

Lu, J., Zhou, X.S., 2023. Virtual track networks: A hierarchical modeling framework and open-source tools for simplified and efficient connected and automated mobility (CAM) system design based on general modeling network specification (GMNS). Transp. Res. Part C: Emerg. Technol. 153, 104223.

Mahmoudi, M., Tong, L.C., Garikapati, V.M., Pendyala, R.M., Zhou, X., 2021. How many trip requests could we support? An activity-travel based vehicle scheduling approach. Transp. Res. Part C: Emerg. Technol. 128, 103222.

Najmi, A., Rey, D., Waller, S.T., Rashidi, T.H., 2020. Model formulation and calibration procedure for integrated multi-modal activity routing and network assignment models. Transp. Res. Part C: Emerg. Technol. 121, 102853.

Nedic, A., Ozdaglar, A., 2009. Distributed subgradient methods for multi agent optimization. IEEE Trans. Autom. Control 54 (1), 48 61.

Olah, C., 2015. Calculus on computational graphs: Backpropagation in this link.

Oppenheim, N., 1995. Urban Travel Demand Modeling. Wiley, New York.

Ortuzar, J.D., Willumsen, L.G., 2001. Modelling Transport. Wiley, New York.

Patil, P.N., Ross, K.C., Boyles, S.D., 2021. Convergence behavior for traffic assignment characterization metrics. Transportmetr. A: Transp. Sci. 17 (4), 1244–1271.

Pendyala, R.M., Konduri, K.C., Chiu, Y.C., Hickman, M., Noh, H., Waddell, P., Wang, L., You, D., Gardner, B., 2012. Integrated land use–transport model system with dynamic time-dependent activity–travel microsimulation. Transp. Res. Rec. 2303 (1), 19–27.

Raney, B., Cetin, N., Völlmy, A., Vrtic, M., Axhausen, K., Nagel, K., 2003. An agent-based microsimulation model of Swiss travel: first results. Netw. Spat. Econ. 3 (1), 23–41.

Recht, Ben., 2016. Mates of Costate. http://www.argmin.net/2016/05/18/mates-of-costate/.

Ruszczyński, A., 1989. An augmented Lagrangian decomposition method for block diagonal linear programming problems. Oper. Res. Lett. 8 (5), 287 294.

Ryu, S., Chen, A., Choi, K., 2017. Solving the combined modal split and traffic assignment problem with two types of transit impedance function. Eur. J. Oper. Res. 257 (3), 870–880.

Sahinidis, N.V., 1996. BARON: A general purpose global optimization software package. J. Global Optimi Safikhani Zation 8 (2), 201–205.

Saitz, J., 1999. Newton-Raphson method and fixed-point technique in finite element computation of magnetic field problems in media with hysteresis. IEEE Trans. Magn. 35 (3), 1398–1401.

Sbayti, H., Lu, C.C., Mahmassani, H.S., 2007. Efficient implementation of method of successive averages in simulation-based dynamic traffic assignment models for large-scale network applications. Transp. Res. Rec. 2029 (1), 22–30.

Smith, S., Fong, A., Yang, C., Gardner, B., 2018. TravelWorks Integrated Models (No. DOT-VNTSC-FHWA-18-11). United States. Federal Highway Administration. Office of Planning.

Van Cranenburgh, S., Wang, S., Vij, A., Pereira, F., Walker, J., 2022. Choice modelling in the age of machine learning-Discussion paper. J. Choice Model. 42, 100340.

Verbas, İ.Ö., Mahmassani, H.S., Hyland, M.F., Halat, H., 2016. Integrated mode choice and dynamic traveler assignment in multimodal transit networks: mathematical formulation, solution procedure, and large-scale application. Transp. Res. Rec. 2564 (1), 78–88.

Wang, D., Liao, F., Gao, Z., Rasouli, S., Huang, H.J., 2020a. Tolerance-based column generation for boundedly rational dynamic activity-travel assignment in large-scale networks. Transp. Res. Part E: Logist. Transp. Rev. 141, 102034.

Wang, S., Mo, B., Zhao, J., 2020b. Deep neural networks for choice analysis: architecture design with alternative-specific utility functions. Transp. Res. Part C: Emerg. Technol. 112, 234–251.

Wang, S., Wang, Q., Bailey, N., Zhao, J., 2021. Deep neural networks for choice analysis: a statistical learning theory perspective. Transp. Res. B Methodol. 148, 60–81.

Wright, S., Nocedal, J., 1999. Numerical optimization. Springer Science 35 (67–68), 7.

Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: a forward and backward propagation algorithmic framework on a layered computational graph. Transp. Res. Part C: Emerg. Technol. 96, 321–346.

Xiong, C., Zhou, X., Zhang, L., 2018. AgBM-DTALite: an integrated modelling system of agent-based travel behaviour and transportation network dynamics. Travel Behav. Soc. 12, 141–150.

Xiong, C., Yang, X.T., Zhang, L., Lee, M., Zhou, W., Raqib, M., 2021. An integrated modeling framework for active traffic management and its applications in the Washington, DC area. J. Intell. Transp. Syst. 25 (6), 609–625.

Xu, X., Zockaie, A., Mahmassani, H.S., Halat, H., Verbas, Ö., Hyland, M., Hicks, J., 2017. Schedule consistency for daily activity chains in integrated activity-based dynamic multimodal network assignment. Transp. Res. Rec. 2664 (1), 11–22.

Yan, X., Liu, X., Zhao, X., 2020. Using machine learning for direct demand modeling of ridesourcing services in Chicago. J. Transp. Geogr. 83, 102661.

Yao, J., Chen, A., Ryu, S., Shi, F., 2014. A general unconstrained optimization formulation for the combined distribution and assignment problem. Transp. Res. B Methodol. 59, 137–160.

Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Tong, L.C., Zhou, X., 2019. ADMM-based problem decomposition scheme for vehicle routing problem with time windows. Transp. Res. B Methodol. 129, 156–174.

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

Zhang, L., Yang, D., Ghader, S., Carrion, C., Xiong, C., Rossi, T.F., Barber, C., 2018. An integrated, validated, and applied activity-based dynamic traffic assignment model for the Baltimore-washington region. Transp. Res. Rec. 2672 (51), 45–55.

Zhao, X., Yan, X., Yu, A., Van Hentenryck, P., 2020. Prediction and behavioral analysis of travel mode choice: a comparison of machine learning and logit models. Travel Behav. Soc. 20, 22–35.

Zhou, X., Taylor, J., 2014. DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration.

Zhou, Z., Chen, A., Wong, S.C., 2009. Alternative formulations of a combined trip generation, trip distribution, modal split, and trip assignment model. Eur. J. Oper. Res. 198 (1), 129–138.

Zhou, X., Mahmassani, H.S., Zhang, K., 2008. Dynamic micro-assignment modeling approach for integrated multimodal urban corridor management. Transp. Res. Part C: Emerg. Technol. 16 (2), 167–186.

Zhou, X., Cheng, Q., Wu, X., Li, P., Belezamo, B., Lu, J., Abbasi, M., 2022. A meso-to-macro cross-resolution performance approach for connecting polynomial arrival queue model to volume-delay function with inflow demand-to-capacity ratio. Multimodal Transp. 1 (2), 100017.

Zhu, Z., Xiong, C., Chen, X., He, X., Zhang, L., 2018. Integrating mesoscopic dynamic traffic assignment with agent-based travel behavior models for cumulative land development impact analysis. Transp. Res. Part C: Emerg. Technol. 93, 446–462.