Nested Benders Decomposition for Large-Scale Multi-Follower Bilevel Optimization

Dominic C. Flocco, Philine Schiewe, Steven A. Gabriel

Abstract. We propose a scalable nested Benders decomposition (BD) framework for single-leader, multi-follower bilevel optimization problems. The proposed framework is applicable to bilevel optimization problems in which each follower solves a linear program and is particularly well suited for instances involving a large number of followers. By identifying the upper-level decisions as complicating variables, the method exploits the separable structure of the lower-level problems, which decouple by follower once the upper-level decisions are fixed. The algorithm employs a two-level, nested BD scheme comprised of an outer and an inner decomposition. To accelerate convergence, we incorporate two complementary strategies: (i) a closest Benders cuts scheme to reduce the number of outer iterations; and (ii) parallel computing techniques to solve the inner BD subproblems efficiently. The framework is implemented on a distributed computing architecture with code optimized for high-performance execution on large-scale instances. We demonstrate the practical utility of the approach by applying it to a mixed-integer bilevel formulation of a line-planning problem with integrated passenger routing in public transportation systems, involving over 50 million variables. Numerical experiments indicate significant computational improvements over the monolithic baseline and exhibit strong scaling performance as the number of processors increases.

1. Introduction

Bilevel optimization problems provide a powerful framework for hierarchical decision problems. In the simplest form, these problems are comprised of two levels of decision makers: a leader and a follower. In this setting, the constraints of the upper-level (leader) problem are defined in part by a lower-level (follower), parametric optimization problem. From a game theoretic perspective, such problems correspond to Stackelberg games in which the leader agent acts first, influencing the subsequent response of the follower. The leader must therefore anticipate the follower's optimal reaction when selecting its own decision, resulting in a nested optimization structure that is often challenging to analyze and solve. Consequently, bilevel optimization has become a widely used modeling framework in applications such as energy markets, transportation planning, and other system analysis settings; see, e.g., (Dempe 2002; Gabriel et al. 2012).

In their traditional form, bilevel optimization frameworks focus on the single-leader, single-follower setting. However, more complex hierarchical decision structures have also been studied, including tri-level models (Gabriel et al. 2022; Han et al. 2015), multi-leader, multi-follower formulations (Hu and Ralph 2007; Pang and Fukushima 2005), and single-leader, multi-follower settings (Basilico et al. 2020; Borges et al. 2021). This work focuses on single-leader multi-follower games (SLMFGs), in which multiple followers respond simultaneously to a single leader. Such models arise in a wide rage of application, particularly in customer-oriented settings

Date: May 30, 2026.
Key words and phrases. Bilevel optimization, Benders decomposition, Parallel computing, Public transportation, Mixed-integer programming.

where a set of customers or users respond to a centralized planner or regulator. In these cases, the lower-level problem is naturally modeled as a collection of follower optimization problems, which introduces additional analytical and computational challenges. In particular, such formulations often lead to very high-dimensional, large-scale instances that further compound the classical difficulties associated with bilevel optimization.

1.1. **Single-Leader Multi-Follower Games.** SLMFGs arise in a variety of practical applications, including energy markets, transportation planning, and logistics modeling. In energy markets, such models are often used to represent the response of energy users to regulations or decisions made by a centralized operator. For example, Alipour et al. (2018) employ a SLMFG framework to model combined heat and power (CHP) microgrids, where the leader represents the microgrid operator designing a demand response program and the followers are CHP owners who maximize profits from thermal and electrical energy sales. Askeland et al. (2023) and Bailly et al. (2023) study the impact of exogenous factors – such as grid tariffs – on network development plans using a SLMFG formulation in which grid users act as followers seeking to minimize energy costs. In the context of demand response, Woo and Moon (2025) use a single-leader, multi-follower framework to represent the hierarchical interaction between an independent system operator (ISO), which sets subsidy rates and prices, and multiple electricity users that optimize their energy consumption and resource operations. The SLMFG framework has also been applied more broadly to other infrastructure problems, such as wastewater management (U-tapao et al. 2016).

In transportation and logistics, SLMFGs are used to model the responses of users or customers to regulations or system design decisions made by a central operator. For example, Xi et al. (2024) use a SLMFG framework to model mobility-as-a-service (MaaS) platforms, where the MaaS regulator seeks to maximize profits by optimizing service prices and resource allocation, while travelers respond by adjusting their participation in the MaaS platform to minimize travel costs. Cerulli et al. (2024) leverage a SLMFG formulation to model peer-to-peer logistics platforms for last-mile delivery, where the platform assigns orders to carriers at the upper level and each carrier solves a profit-maximizing tour problem at the lower level to determine which requests to accept. Lei et al. (2020) and Zha et al. (2017) study on-demand ride-sharing systems using SLMFG models in which the leader is the ride-sharing company setting surge prices or optimizing system performance, while users or drivers determine their responses to improve convenience or profitability. Finally, Goerigk and Schmidt (2017) study public transport line planning, where a system planner selects lines at the upper level and passengers solve shortest path problems on the resulting network at the lower level.

1.2. **Multi-Follower Bilevel Optimization Problems.** We consider a general, multi-follower bilevel optimization problem. Denote by $\mathcal{F} = \{1, \dots, n\}$ the set of lower-level players (e.g., followers), each indexed by $f \in \mathcal{F}$. Each follower f has decision vector $y_f \in \mathbb{R}^{n_f}$, and we denote by $y \in \{y_f\}_{f \in \mathcal{F}}$ the vector of all lower-level decisions. The upper-level decisions are denoted by the vector $x \in \mathbb{R}^{N_x}$. Consider

the bilevel optimization problem:

$$\underset{x,y}{\text{minimize}} \quad c_x^\top x + \sum_{f=1}^{n} c_f^\top y_f \tag{1a}$$

$$\text{s.t.} \quad Ax + \sum_{f=1}^{n} B_f y_f \geq a \tag{1b}$$

$$y_f \in \underset{\bar{y}_f \in \mathbb{R}^{n_f}}{\arg\min} \left\{ d_f^\top \bar{y}_f : C_f x + D_f \bar{y}_f \geq b_f \right\} \qquad \forall f \in \mathcal{F} \tag{1c}$$

$$x \in \mathcal{X} \tag{1d}$$

We note that the specific definition of $\mathcal{X}$ may vary by application and ultimately affects the computational difficulty of solving the master problem; for example, x may be constrained to be integer, binary, or purely continuous, or any combination thereof. Note that the lower-level problems (1c) assume continuous variables only. Furthermore, we denote by $\lambda_f \in \mathbb{R}^{m_f}$ the Lagrange multiplier vector associated with the inequality constraints in (1c).

Solving linear bilevel problems of the form (1) is known to be NP-hard, even in the single-follower case (Bard 1991). The principle challenge lies in the hierarchical structure of the lower-level problem (1c). Typical approaches reformulate (1c) to arrive at an equivalent single-level reformulation. The most common approach is to replace (1c) with its Karush-Kuhn-Tucker (KKT) conditions, leading to a mathematical program with equilibrium constraints (MPEC) (Luo et al. 1996). The complementarity slackness conditions $\lambda_f^\top (C_f x + D_f y_f) = 0$ can then be reformulated using a big-M constraint, creating a mixed-integer linear program (MILP) (Fortuny-Amat and McCarl 1981), or via Special-Order Sets of type 1 (Siddiqui and Gabriel 2013). There exists an expansive literature on alternative solution methods and reformulation techniques for the linear bilevel problem (1); see, e.g., (Bard 1998; Dempe et al. 2015; Mitsos et al. 2008).

The MPEC reformulation of linear bilevel problems can require binary variables for each complementarity slackness condition, resulting in poor scaling performance. Moreover, (Kleinert et al. 2020) show that finding the correct big-M for linear bilevel problems is NP-hard in its own right. To address these drawbacks, a common approach is to employ the strong-duality property of linear programs to reformulate (1) using the a primal-dual inequality for each follower (Beheshti et al. 2016; Dimanchev et al. 2024; Huppmann and Egerer 2015). These constraints contain bilinear terms of the form $\lambda_f^\top C_f x$, resulting in a quadratically constrained optimization problem with a nonconvex feasible region for the single-level formulation. Zare et al. (2019) show that strong-duality-based approaches can significantly reduce computational times compared to KKT-based ones. In this work, we adopt the strong-duality reformulation to solve (1) and develop a tailored Benders decomposition that exploits the structure of leader-follower interactions to efficiently solve the problem.

1.3. **Summary of Contribution.** We propose a parallel nested Benders decomposition (BD) algorithm for solving bilevel optimization problems of the form (1). The proposed framework is applicable to bilevel optimization problems in which each follower solves a linear program and is particularly well suited for instances involving a large number of followers. Along these lines, we apply the approach to the line-planning problem with integrated passenger routing introduced by (Schöbel and Scholl 2006). The method exploits the identification of the upper-level decision variables as complicating and the separable structure of the lower-level problems, which decompose by follower once the upper-level decisions are fixed. The algorithm

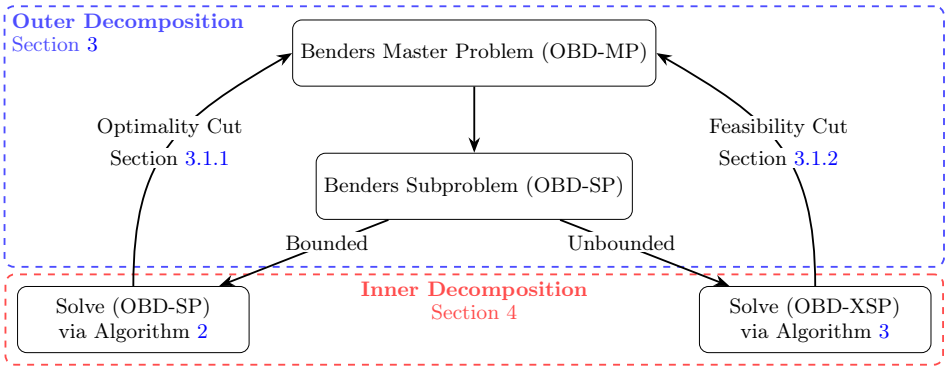


FIGURE 1. Visualization of Nested Benders Decomposition Scheme

applies a nested, two-level BD scheme, consisting of an outer and an inner decomposition. The algorithm incorporates two complementary strategies to enhance Benders convergence in this setting: (1) the use of closest Benders cuts, following Seo et al. (2022), to reduce the number of outer iterations required for convergence; and (2) the efficient solution of the inner BD subproblems through parallel computing techniques. To the best of our knowledge, this is the first nested Benders decomposition algorithm explicitly designed for parallel execution on modern distributed-memory architectures.

Figure 1 provides a high-level roadmap of the nested Benders scheme applied to the single-level reformulation of the bilevel problem (7). The central idea is to separate the problem into an outer decomposition, which handles the complicating variables (i.e., the upper-level variables, highlighted in the blue box), and an inner decomposition for the resulting subproblem. In the outer Benders decomposition (OBD) subproblem, the complicating constraints give rise to dual variables in the corresponding dual subproblem. This dual subproblem, which contains complicating variables, is then solved using the inner decomposition, highlighted in the red box. The inner scheme performs a classical Benders decomposition on the subproblem of the outer iteration: if the outer subproblem is bounded, the inner scheme generates optimality cuts; if unbounded, it generates feasibility cuts, which are sent back to the master problem. The advantage of this approach is that the inner subproblem becomes fully decomposable by follower and can be solved in parallel for improved computational efficiency.

1.4. **Paper Organization.** The remainder of the paper is organized as follows. Section 2 presents preliminaries on BD and reviews recent enhancements to the classical approach. Sections 3 and 4 develop the nested BD framework by first presenting the outer decomposition scheme, followed by the inner scheme, and illustrating the nested relationship between the two. Section 5 describes the parallel implementation of the nested BD framework on a distributed computing model. Section 6 presents numerical experiments for the SLMFG application in public transportation planning. Finally, Section 7 concludes the paper and discusses directions for future research.

2. BACKGROUND

The Benders decomposition algorithm was originally proposed by Benders (1962), and has since been widely developed as an exact algorithm for solving mixed integer programming problems. It has been successfully applied to various applications in operations research and systems engineering, including network design problems

(Fortz and Poss 2009), transportation problems (Gelareh et al. 2015), and scheduling problems (Mercier and Soumis 2007). In this section, we present the classical version of the Benders algorithm and survey relevant literature on the method, with a particular focus on nested implementations and application to equilibrium problems. For a comprehensive survey of Benders decomposition and its modern variants, including recent algorithmic advancements and acceleration techniques, the reader is referred to (Rahmaniani et al. 2017).

2.1. **Classical Benders Decomposition.** The classical Benders decomposition algorithm considers an optimization problem of the form:

$$\underset{x,y}{\text{minimize}} \quad c_x^\top x + c_y^\top y \tag{2a}$$

$$\text{s.t.} \quad Ax + By \geq a \tag{2b}$$

$$x \in \mathcal{X}, y \in \mathbb{R}_+^{N_y}, \tag{2c}$$

where $c_x \in \mathbb{R}^{N_x}$, $c_y \in \mathbb{R}^{N_y}$, $A \in \mathbb{R}^{M \times N_x}$, $B \in \mathbb{R}^{M \times N_y}$ and $a \in \mathbb{R}^M$. We denote by $\mathcal{X} \subset \mathbb{R}^{N_x}$ the feasible set of the complicating variable x. In conventional applications, the set $\mathcal{X}$ typically contains integer polyhedral constraints, e.g., $\mathcal{X} = \{x \in \mathbb{Z}_+^{N_x} : Wx \geq q\}$ for $W \in \mathbb{R}^{\ell \times N_x}$ and $q \in \mathbb{R}^\ell$. The optimization problem (2) can be written as

$$\underset{x \in \mathcal{X}}{\text{minimize}} \; c_x^\top x + \min_{y \in \mathbb{R}_+^{N_y}} \{c_y^\top y : By \geq a - Ax\}. \tag{3}$$

Let $\pi \in \mathbb{R}^M$ be the dual vector associated with the constraints $By \geq a - Ax$. The Benders subproblem is the dual of the inner minimization. For a fixed $x^* \in \mathcal{X}$ the subproblem is expressed as

$$\underset{\pi \in \mathbb{R}_+^M}{\text{maximize}} \quad \pi^\top (a - Ax^*) \quad \text{s.t.} \quad \pi^\top B \leq c_y \tag{4}$$

The feasible region $\Omega = \{\pi \in \mathbb{R}_+^M : \pi^\top B \leq c_y\}$ of (4) is independent of the choice of x^*; thus, the subproblem may be unbounded for an arbitrary choice of $x^* \in \mathcal{X}$, indicating infeasibility of x^* w.r.t. the inner minimization in (3). In this case, given the set of extreme rays Ω^R of Ω, there is a direction of unboundedness $\pi_r \in \Omega^R$ for which $\pi_r^\top (a - Ax^*) > 0$. To avoid this, we add a constraint $\pi_r^\top (a - Ax^*) \leq 0$ for all $\pi_r \in \Omega^R$ to the subproblem (4), which are referred to as feasibility cuts. In practice, the extreme rays of Ω^R will be identified by examining directions of unboundedness.

If the subproblem (4) is bounded for some choice of x^*, then there exists a solution at one of the extreme points $\pi_e \in \Omega^E$, where Ω^E is the set of extreme points of Ω. Note that other solutions may also exists as convex combinations of these extreme points. Therefore, including the feasibility cuts as well, the subproblem (4) can be reformulated as

$$\underset{e \in \Omega^E}{\text{maximize}} \quad \pi_e^\top (a - Ax^*) \quad \text{s.t.} \quad \pi_r^\top (a - Ax^*) \leq 0 \; \forall \pi_r \in \Omega^R, \tag{5}$$

which can be linearized via the introduction of auxiliary variable $\theta \in \mathbb{R}$ to give the Benders master problem:

$$\underset{x,\theta}{\text{minimize}} \quad c_x^\top x + \theta \tag{6a}$$

$$\text{s.t.} \quad \pi_e^\top (a - Ax) \leq \theta \quad \forall \pi_e \in \Omega^E \tag{6b}$$

$$\pi_r^\top (a - Ax) \leq 0 \quad \forall \pi_r \in \Omega^R \tag{6c}$$

$$x \in \mathcal{X}. \tag{6d}$$

Constraints (6b) and (6c) are referred to as optimality cuts and feasibility cuts, respectively. Under exact enumeration of the sets Ω^E and Ω^R, this problem is equivalent to (2). However, the size of these sets, and therefore the number of constraints, can be exponentially large, making it impractical to consider all feasibility and optimality cuts. Moreover, these sets are not generally known in closed form *a priori*. Therefore, Benders (1962) proposed a cutting plane method, which iteratively solves the master problem (6) with only a subset of constraints (6b) and (6c). The result is an algorithm that repeatedly solves (6) to obtain a candidate solution x^*, then solves the subproblem (4) given this candidate x^*. If the subproblem is unbounded, a feasibility cut of the form (6c) is produced. If the subproblem is feasible, an optimality cut of the form (6b) is produced. At each iteration, the generated optimality and feasibility cuts are added to the master problem, dynamically generating the sets Ω^E and Ω^R. In the context of two-stage stochastic programming, this iterative cutting-plane procedure is known as the L-shaped method, a special case of Benders decomposition introduced by (Van Slyke and Wets 1969).

2.2. **Literature Review.** The BD algorithm was originally developed for a class of mixed-integer linear programs (MILPs), where a subset of integer variables is complicating and, when fixed, yield a continuous linear program amenable to classical duality theory for cut generation. The framework has since been extended to a much broader family of problems exhibiting decomposable structure. Most notably for this work, BD has been successfully applied to large-scale nonconvex mixed integer nonlinear programs (Lin and Üster 2014; Sahinidis and Grossmann 1991), multi-stage stochastic problems (Adulyasak et al. 2015; Linderoth and Wright 2003; Wolf 2014) and bilinearly constrained programs (Fontaine and Minner 2014), among others. Beyond these settings, BD has also been employed in equilibrium contexts, specifically for complementarity problems, and, more generally, variational inequalities, where it has proven effective for computing equilibria, see, e.g., (Fuller and Chung 2008; Gabriel and Fuller 2010; Luna et al. 2020). A unifying theme across these applications is the identification of complicating variables and the systematic exploitation of problem structure to enable an efficient decomposition.

Considerable research has focused on improving the convergence of the BD algorithm by reducing the number of iterations required for convergence and the computational effort per iteration. Rahmaniani et al. (2017) identifies four key research directions for improving the computational performance of Benders decomposition: (1) decomposition strategy, (2) solution generation, (3) solution procedure, and (4) cut generation. In this work, we focus on directions (3) and (4) by improving the efficiency of subproblem solution procedures, and by developing stronger cuts that yield higher-quality candidate solutions. Specifically, we incorporate parallelization strategies and close cuts to address these aspects. Here, we survey the relevant literature on these two research directions and refer the reader to (Rahmaniani et al. 2017) and the references therein for a comprehensive review of classic and modern BD literature.

The subproblem (4) may be large or further decomposable into smaller subproblems; various strategies have been proposed to solve the subproblem more effectively. Zakeri et al. (2000) show that suboptimal solutions of the dual subproblem can be used to generate useful valid cuts, which are computationally less expensive and produce good results. When subproblems have a decomposable structure, an inner decomposition such as column generation or BD can be employed on the subproblem to yield a nested procedure, such as in (Cordeau et al. 2001; Mercier and Soumis 2007; Papadakos 2009). When there are many, separable subproblems, parallel computing techniques can be useful. In such techniques, one processor solves the master problem and coordinates with other processors to solve the subproblems.

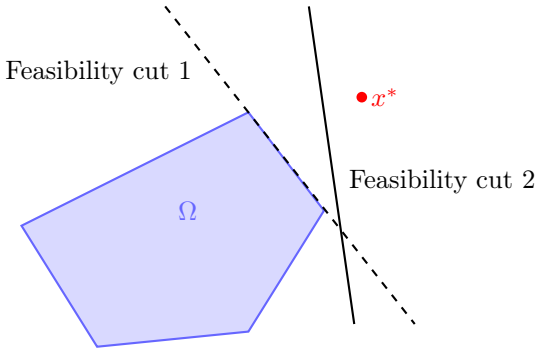


FIGURE 2. Strong feasibility cut given candidate solution x^* and feasible region Ω.

Experimentation has shown this strategy effective, namely (Fontaine and Minner 2014; Linderoth and Wright 2003; Vladimirou 1998; Wolf and Koberstein 2013). In these methods, the distribution of subproblems on processors has important effects on performance; specifically, Chermakani (2015) observe that when the number of subproblems is considerably larger than the number of available processors, it may be better to aggregate some of the subproblems. In this work, we present a nested Benders algorithm, where the subproblems of the inner Benders decomposition can be solved in parallel.

Benders cut generation concerns the strategy used to generate optimality and feasibility cuts. The number of iterations is closely related to the strength of the cuts, i.e., the values selected for the dual variables π in (6b) and (6c). Fischetti et al. (2010) propose a new selection criterion for Benders cuts based on solving a certain cut generation LP, where cut violation act as the objective function to be maximized subject to a normalization condition. This concept is based on the fact that finding the most-violated optimality cut is equivalent to finding an optimal vertex of a polyhedron with unbounded rays (Fukuda and Prodon 1996). Yang and Lee (2012) build upon this approach to generate feasibility cuts corresponding to faces of a polyhedron defined by all feasibility cuts. Recently, Seo et al. (2022) extend the approach of Yang and Lee (2012) to build a unified framework to generate the closest cut among all feasibility and optimality cuts efficiently. Their closest Benders cuts approach selects cuts based on their geometric proximity to the current feasible solution, i.e., the cut whose hyperplane lies nearest to the feasible point, similar to (Hosseini and Turner 2024), a concept also referred to as cut depth. The idea of a strong, or closest, cut is illustrated in Figure 2, where cut 1 is geometrically closer to the feasible region Ω, i.e., stronger, than cut 2. The proposed nested Benders decomposition scheme presented in this paper utilizes the closest cut selection procedure from Seo et al. (2022) to decrease the number of outer Benders iterations required for convergence.

3. Outer Benders Decomposition

To solve (1), we recast the bilevel problem as a single-level optimization problem using strong duality instead of complementarity. The notation used to describe this reformulation and the nested Benders scheme to follow are outlined in Table 1. The strong duality approach follows from the work by Huppmann and Egerer (2015), which is successfully applied to a class of trilevel optimization problems in (Herrala et al. 2025) and power market equilibrium problems in (Dimanchev et al. 2024). The primary benefit of this approach is that it avoids many of the bilinear constraints

Symbol	Dimension	Description
		Dimensions
n	-	Number of followers $\lvert\mathcal{F}\rvert$
n_f	-	Dimension of follower f's decision vector
m_f	-	Number of constraints in follower f's problem (1c)
N_y	-	Number of lower-level decisions: $\sum_{f=1}^{n} n_f$
M_y	-	Number of lower-level constraints: $\sum_{f=1}^{n} m_f$
N_x	-	Dimension of leader's decision vector
M_x	-	Number of constraints in upper-level (1b)
		Problem Data
c_x	N_x	Leader's upper-level objective coefficients
a	M_x	RHS of upper-level constraints
c_f	n_f	Follower f's upper-level objective coefficients
d_f	n_f	Follower f's lower-level objective coefficients
b_f	m_f	RHS of follower f's lower-level constrains
A	$M_x \times N_x$	Leader's upper-level constraint matrix
B_f	$M_x \times n_f$	Follower f's upper-level constraint matrix
C_f	$m_f \times N_x$	Leader's constraint matrix in follower f's problem
D_f	$m_f \times n_f$	Follower f's lower-level constraint matrix
		Problem Variables in (7)
x	n_x	Leader's decision vector
y_f	n_f	Follower f's decision vector
λ_f	m_f	Dual multiplier to follower f's primal constraints
		BD Algorithm Variables
μ	M_x	Dual multiplier for upper-level feasibility constraint (7b)
π_f	m_f	Dual multiplier to follower f's primal feasibility constraints (7c)
ξ_f	n_f	Dual multiplier to follower f's dual feasibility constraints (7d)
ζ_f	1	Dual multplier to follower f's strong duality constraint (7e)

TABLE 1. Summary of Notation

that would otherwise arise from complementary slackness conditions. In particular, this approach replaces the complementarity constraints in the KKT conditions of each follower's optimization problem (1c) with a strong duality constraint, resulting in the following single-level reformulation:

$$\text{(BL)} \quad \underset{x,y,\lambda}{\text{minimize}} \quad c_x^\top x + \sum_{f=1}^{n} c_f^\top y_f \tag{7a}$$

$$\text{s.t.} \quad Ax + \sum_{f=1}^{n} B_f y_f \geq a \qquad (\mu) \tag{7b}$$

$$C_f x + D_f y_f \geq b_f \qquad (\pi_f) \qquad \forall f \in \mathcal{F} \tag{7c}$$

$$D_f^\top \lambda_f = d_f \qquad (\xi_f) \qquad \forall f \in \mathcal{F} \tag{7d}$$

$$d_f^\top y_f \leq \lambda_f^\top (b_f - C_f x) \qquad (\zeta_f) \qquad \forall f \in \mathcal{F} \tag{7e}$$

$$x \in \mathcal{X}, \tag{7f}$$

where $\mu \in \mathbb{R}^{M_x}$, $\pi_f \in \mathbb{R}^{m_f}$, $\xi_f \in \mathbb{R}^{n_f}$ and $\zeta_f \in \mathbb{R}$ for $f \in \mathcal{F}$ are Lagrange multipliers for the respective constraints. Constraints (7c) and (7d) enforce primal and dual feasibility for each follower's optimization problem. The reverse weak duality constraint (7e) requires that the objective value of each follower's primal

problem be no greater than that of the corresponding dual problem. By the weak duality theorem (Bertsimas and Tsitsiklis 1997), the objective value of any primal (minimization) feasible solution is greater than or equal to the objective value of any dual (maximization) feasible solution, which allows the strong duality condition to be expressed in the inequality form given in (7e). Considering this strong duality constraint as an inequality also enlarges the feasible region of the single-level reformulation, which can improve computational performance. A remaining computational difficulty arises from the quadratic term $\lambda_f^\top C_f x$ in (7e). However, when the upper-level decision variable x is fixed, the resulting problem reduces to a continuous linear program. Consequently, x acts as the complicating variable, which we address using Benders decomposition.

The proposed nested BD algorithm is based on the observation that, given a fixed upper-level decision x, (7) is nearly decomposable by follower f, with the exception of the complicating constraints (7b). Specifically, constraints (7b) and (7c) can be written as the following $(M_x + M_y) \times (N_x + N_y)$ linear system:

$$\begin{bmatrix} A & B_1 & B_2 & \cdots & B_n \\ C_1 & D_1 & 0 & \cdots & 0 \\ C_2 & 0 & D_2 & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ C_n & 0 & \cdots & \cdots & D_n \end{bmatrix} \begin{bmatrix} x \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \geq \begin{bmatrix} a \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}. \tag{8}$$

Moreover, the lower-level dual feasibility constraints (7d) can be written as a $N_y \times M_y$ block-diagonal linear system:

$$\operatorname{diag}(D_1^\top, \ldots, D_n^\top)\lambda = d. \tag{9}$$

Also, the "reverse weak duality" constraints (7e) are separable by each follower f if x is fixed. In light of (8), the single-level reformulation (7) of the bilevel problem (1) contains both complicating variables and complicating constraints. The upper-level decision variables $x \in \mathcal{X}$ serve as the complicating variables, while the linking constraints (7b) constitute the complicating constraints. The central idea of the proposed approach is to apply a nested decomposition scheme to solve (7), in which the outer iteration handles the complicating variables x, while the inner iterations address the complicating constraints. To address problems with complicating constraints, the Dantzig–Wolfe decomposition is the conventional approach; see, e.g., (Barnhart et al. 1998; Dantzig and Wolfe 1960; Fuller and Chung 2005). However, we consider the dual of the outer subproblem, in which the Lagrange multipliers associated with the complicating constraints become complicating variables. Consequently, the inner decomposition scheme is also a Benders decomposition.

The outer Benders decomposition (OBD) applies BD to address the complicating upper-level decision x. To apply BD to (7), we first reformulate the problem as follows:

$$\underset{x \in \mathcal{X}}{\text{minimize}} \quad c_x^\top x + \theta(x), \tag{10}$$

where

$$\theta(x) = \min_{y,\lambda} \quad \sum_{f=1}^{n} c_f^\top y_f \tag{11a}$$

$$\text{s.t.} \quad \sum_{f=1}^{n} B_f y_f \geq a - Ax \tag{11b}$$

$$D_f y_f \geq b_f - C_f x \qquad \forall f \in \mathcal{F} \tag{11c}$$

$$D_f^\top \lambda_f = d_f \qquad \forall f \in \mathcal{F} \tag{11d}$$

$$(b_f - C_f x)^\top \lambda_f - d_f^\top y_f \geq 0 \qquad \forall f \in \mathcal{F}. \tag{11e}$$

The value function $\theta(x)$ is piecewise linear and convex since (11) is a linear program given a fixed x (Conejo et al. 2006). Given a fixed value x^*, the outer Benders subproblem is given by

$$\underset{y,\lambda}{\text{minimize}} \quad \sum_{f=1}^{n} c_f^\top y_f \tag{12a}$$

$$\text{s.t.} \quad \sum_{f=1}^{n} B_f y_f \geq a - Ax^* \qquad (\mu) \tag{12b}$$

$$D_f y_f \geq b_f - C_f x^* \qquad (\pi_f) \quad \forall f \in \mathcal{F} \tag{12c}$$

$$D_f^\top \lambda_f = d_f \qquad (\xi_f) \quad \forall f \in \mathcal{F} \tag{12d}$$

$$(b_f - C_f x^*)^\top \lambda_f - d_f^\top y_f \geq 0 \qquad (\zeta_f) \quad \forall f \in \mathcal{F} \tag{12e}$$

$$\lambda_f \geq 0,\ y_f \text{ free} \qquad \forall f \in \mathcal{F} \tag{12f}$$

The dual of this subproblem is given by

$$\text{(OBD-SP)} \ \underset{\mu,\pi,\xi,\zeta}{\text{maximize}} \quad \mu^\top (a - Ax^*) + \sum_{f=1}^{n} \pi_f^\top (b_f - C_f x^*) + \xi_f^\top d_f \tag{13a}$$

$$\text{s.t.} \quad B_f^\top \mu + D_f^\top \pi_f - d_f \zeta_f = c_f \qquad (y_f) \quad \forall f \in \mathcal{F} \tag{13b}$$

$$D_f \xi_f + (b_f - C_f x^*)\zeta_f \leq 0 \qquad (\lambda_f) \quad \forall f \in \mathcal{F} \tag{13c}$$

$$\mu, \pi, \zeta \geq 0,\ \xi \text{ free} \tag{13d}$$

Observe that the complicating constraints (12b) in the primal subproblem are dualized and thus appear as the dual variable μ in the dual subproblem (13). As a result, the source of complication in solving the primal (12) is transferred from the constraints (12b) to the associated dual variables μ. This transformation will ultimately allow us to apply an inner BD when solving the outer subproblem (13).

For a fixed candidate solution x^*, define $\Omega^E(x^*)$ to be the set of extreme points of the feasible region (13) and $\Omega^R(x^*)$ the set of extreme rays of the recession cone

$$\Omega^R(x^*) = \left\{ \begin{array}{l} (\mu, \pi, \xi, \zeta) : B_f^\top \mu + D_f^\top \pi_f - d_f \zeta_f = 0,\ \forall f \in \mathcal{F} \\ \qquad\qquad D_f \xi_f + (b_f - C_f x^*)\zeta_f \leq 0,\ \forall f \in \mathcal{F} \\ \qquad\qquad \mu, \pi, \zeta \geq 0,\ \xi \text{ free} \end{array} \right\}, \tag{14}$$

which represents the set of unbounded directions of the feasible region of the dual subproblem (13). Formally, it characterizes the directions in which one can move indefinitely away from any feasible point without leaving the feasible set. In the context of (13b), let $\omega^0 = (\mu^0, \pi_f^0, \xi_f^0, \zeta_f^0)$ be a feasible point and $\omega^r = (\mu^r, \pi_f^r, \xi_f^r, \zeta_f^r)$ be a direction. For ω^r to be in the recession cone, i.e., a direction of unboundedness,

the point $\omega^0 + \alpha\omega^r$ must be feasible for all $\alpha \geq 0$, i.e.,

$$\begin{aligned} & B_f^\top(\mu^0 + \alpha\mu^r) + D_f^\top(\pi_f^0 + \alpha\pi_f^r) - d_f(\zeta_f^0 + \alpha\zeta_f^r) = c_f \\ \iff & \underbrace{B_f^\top\mu^0 + D_f^\top\pi_f^0 - d_f\zeta_f^0}_{=c_f} + \alpha\left[B_f^\top\mu^r + D_f^\top\pi_f^r + d_f\zeta_f^r\right] = c_f, \end{aligned} \tag{15}$$

and

$$\begin{aligned} & D_f(\xi_f^0 + \alpha\xi_f^r) + (b_f - C_f x^*)(\zeta_f^0 + \alpha\zeta_f^r) \leq 0 \\ \iff & \underbrace{D_f\xi_f^0 + (b_f - C_f x^*)\zeta_f^0}_{\leq 0} + \alpha\left[D_f\xi_f^r + (b_f - C_f x^*)\zeta_f^r)\right] \leq 0. \end{aligned} \tag{16}$$

Take $\alpha \to \infty$. Looking first at (15), if $B_f^\top\mu^r + D_f^\top\pi_f^r + d_f\zeta_f^r \neq 0$, then any value of $\alpha > 0$ would immediately violate the equality, making the solution infeasible. For (16), if $D_f\xi_f^r + (b_f - C_f x^*)\zeta_f^r > 0$, then the left-hand-side will eventually become greater than 0. Alternatively, if $D_f\xi_f^r + (b_f - C_f x^*)\zeta_f^r \leq 0$, then the left-hand-side will decrease without bound as $\alpha \to \infty$, indicating a direction of unboundedness. Together, we arrive at the recession cone $\Omega^R(x^*)$ given by (14), which represents the set of unbounded directions.

Given the current candidate solution x^*, the outer Benders master problem is then

(OBD-MP)

$$\underset{x,\theta}{\text{minimize}} \quad c_x^\top x + \theta \tag{17a}$$

$$\text{s.t.} \quad (\bar{\mu}^r)^\top(a - Ax) + \sum_{f=1}^{n}(\bar{\pi}_f^r)^\top(b_f - C_f x) + (\bar{\xi}_f^r)^\top d_f \leq 0 \qquad \forall(\bar{\mu}^r, \bar{\pi}^r, \bar{\xi}^r) \in \Omega^R(x^*) \tag{17b}$$

$$(\mu^e)^\top(a - Ax) + \sum_{f=1}^{n}(\pi_f^e)^\top(b_f - C_f x) + (\xi_f^e)^\top d_f \leq \theta \qquad \forall(\mu^e, \pi^e, \xi^e) \in \Omega^E(x^*) \tag{17c}$$

$$x \in \mathcal{X}, \tag{17d}$$

where constraints (17b) and (17c) represent the feasibility and optimality cuts of the outer BD loop, respectively. While ζ_f is a necessary component of the dual feasible region (13b)-(13c), it is associated with the primal constraint (12e) which has a zero right-hand side. Consequently, ζ_f does not appear in the cut expressions (17b) and (17c). To generate the cuts (17b) and (17c), we solve (13) using an inner Benders decomposition scheme. We first describe the construction of close cuts and then explain how the inner decomposition is applied to obtain solutions to (13) that inform both feasibility and optimality cuts.

3.1. **Benders Cut Generation.**

To generate feasibility and optimality cuts, we adopt the closest-cut procedure proposed by (Seo et al. 2022). This approach leverages the geometric interpretation of cut depth to provide a unified framework for selecting the "closest" cut among all feasibility and optimality cuts. Specifically, the scheme determines how the unbounded dual rays $\bar{\mu}^r$, $\bar{\pi}_f^r$ and $\bar{\xi}_f^r$ or the extreme points μ^e, π_f^e and ξ_f^e, are computed for feasibility cuts (17b) and optimality cuts (17c), respectively. For each case, i.e., feasibility and optimality cut generation, we replace the subproblem (13) with an alternative problem formulation and solution scheme.

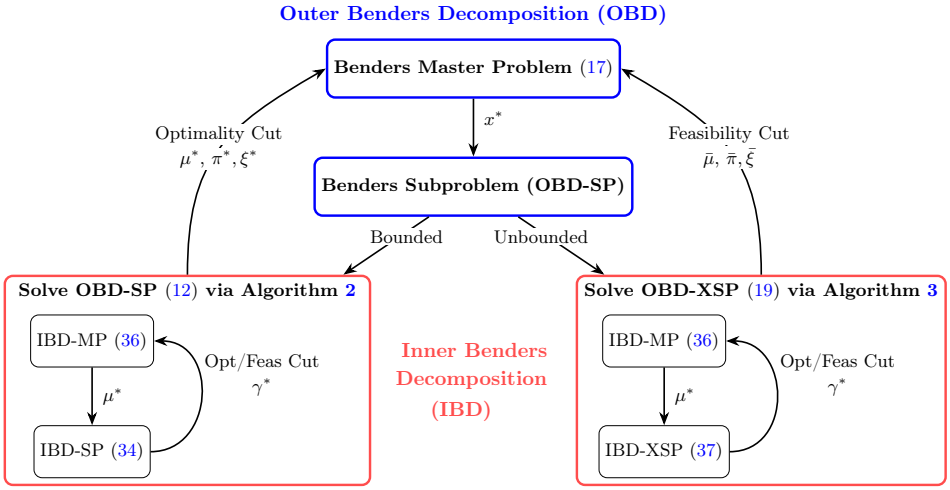


FIGURE 3. Nested Benders Decomposition Scheme

At each iteration, we determine whether feasibility or optimality cuts should be generated by checking whether the candidate solution x^*, obtained from OBD-MP (17), is feasible for the primal subproblem (12) or, equivalently, whether the dual subproblem (13) is bounded. If (13) is bounded at x^*, we generate optimality cuts as described in Section 3.1.1. Otherwise, if (13) is unbounded at x^*, we generate feasibility cuts as described in Section 3.1.2.

3.1.1. *Selecting Optimality Cuts.* First, we describe the optimality cut selection procedure, which is more straightforward than the feasibility cut routine to be described in Section 3.1.2. When generating an optimality cut, we have a candidate solution x^* that is feasible for (12), meaning that (13) is bounded. In this case, we solve the dual problem OBD-SP (13) to obtain an extreme point $(\mu^e, \pi^e, \xi^e, \zeta^e) \in \Omega^E(x^*)$, which is then used to construct optimality cuts of the form (17c). Note that (13) is decomposable with μ as the complicating variable, allowing it to be solved efficiently using an inner Benders decomposition scheme, described in the next section.

3.1.2. *Selecting Feasibility Cuts.* If the primal Benders subproblem (13) is unbounded, we must add feasibility cuts to the master problem OBD-MP (17). These feasibility cuts eliminate primal-infeasible directions in the master problem (17). Specifically, we seek to generate the unbounded rays $\bar{\omega} = (\bar{\mu}, \bar{\pi}, \bar{\xi}, \bar{\zeta})$ of the Benders subproblem (13). These rays can be computed by solving the following problem (Seo et al. 2022):

$$z_{feas}(x^*) = \max_{\mu,\pi,\zeta,\xi} \quad \mu^\top (a - Ax^*) + \sum_{f=1}^{n} \pi_f^\top (b_f - Cx^*) + \xi_f^\top d_f \tag{18a}$$

$$\text{s.t.} \quad B_f^\top \mu + D_f^\top \pi_f - \zeta_f d_f = 0 \qquad \forall f \in \mathcal{F} \tag{18b}$$

$$D_f \xi_f + (b_f - C_f x^*)\zeta_f \leq 0 \qquad \forall f \in \mathcal{F} \tag{18c}$$

$$\mathbb{1}^\top \mu + \sum_{f=1}^{n} \left(\mathbb{1}^\top \pi_f + \|\xi_f\|_1 + \zeta_f\right) = 1 \tag{18d}$$

$$\mu, \pi, \zeta \geq 0, \ \xi \text{ free.} \tag{18e}$$

Here, $\mathbb{1}$ denotes the vector of ones. Constraints (18b)–(18c) ensure that all feasible solutions to (18) lie in the recession cone $\Omega^R(x^*)$ of the feasible region of (13),

while (18d) imposes normalization. Consequently, any solution $(\bar{\mu}, \bar{\pi}, \bar{\xi}, \bar{\zeta})$ to (18) corresponds to a normalized unbounded direction of (13).

If $z_{feas}(x^*) > 0$, then (13) is unbounded, implying that the primal subproblem (12) is infeasible. In this case, the resulting extreme ray yields a feasibility cut of the form (17b). The intuition behind why these rays produce strong feasibility cuts is illustrated in Example 1 (below); further discussion is provided in (Seo et al. 2022).

We would like (18) to be decomposable by follower f, with complications arising only from the variables μ so that the inner Benders decomposition can be applied to solve it. However, the normalization constraint (18d) introduces an additional complication in the form of a linking constraint that prevents the subproblem from being decomposed in this manner. Any solution to (18) without the normalization constraint (18d) still corresponds to a recession direction, but it may not yield strong feasibility cuts in the absence of normalization. To address this, we propose the following approach. First, solve a relaxed version of (18) without the normalization constraint (18d) removed:

$$\begin{aligned}
\text{(OBD-XSP)} \quad \underset{\mu,\pi}{\text{maximize}} \quad & \mu^\top (a - Ax^*) + \sum_{f=1}^{n} \pi_f^\top (b_f - Cx^*) + \xi_f^\top d_f && \text{(19a)}\\
\text{s.t.} \quad & B_f^\top \mu + D_f^\top \pi_f - \zeta_f d_f = 0 \qquad \forall f \in \mathcal{F} && \text{(19b)}\\
& D_f \xi_f + (b_f - C_f x^*)\zeta_f \leq 0 \qquad \forall f \in \mathcal{F} && \text{(19c)}\\
& \mu \in [0,1]^{M_x}, \pi \in [0,1]^{M_y}, && \text{(19d)}\\
& \zeta \in [0,1]^n, \xi \in [-1,1]^{N_y}, && \text{(19e)}
\end{aligned}$$

where $\pi = \{\pi_f\}_{f=1}^n$, $\xi = \{\xi_f\}_{f=1}^n$ and $\zeta = \{\zeta_f\}_{f=1}^n$. This problem can be solved via Benders decomposition with complicating variables μ. The following results establish that any solution to OBD-XSP (19) yields a valid feasibility cut of the form (17b).

Lemma 3.1. *Let x^* be a candidate solution computed by OBD-MP (17). If the subproblem (12) is infeasible at x^*, then for any recession direction $\bar{\omega} = (\bar{\mu}, \bar{\pi}, \bar{\xi}, \bar{\zeta}) \in \Omega^R(x^*)$, the inequality (17b) is a valid feasibility cut for (7). In particular, this cut is satisfied by every feasible solution x of (7).*

Proof. Since the primal subproblem (12) is infeasible at x^*, then the dual OBD-SP (13) must be unbounded. By Farkas' Lemma, there exists a recession direction $\bar{\omega} = (\bar{\mu}, \bar{\pi}, \bar{\xi}, \bar{\zeta}) \in \Omega^R(x^*)$ such that

$$\bar{\mu}^\top (a - Ax^*) + \sum_{f=1}^{n} \bar{\pi}_f^\top (b_f - C_f x^*) + \bar{\xi}_f^\top d_f > 0. \tag{20}$$

Let (x, y, λ) be any feasible solution to the original problem (7). We will show that x satisfies the feasibility cut (17b) for $(\bar{\mu}, \bar{\pi}, \bar{\xi})$. From the primal constraint (7b) and the fact that $\bar{\mu} \geq 0$, we have:

$$\sum_{f=1}^{n} \bar{\mu}^\top B_f y_f \geq \bar{\mu}^\top (a - Ax). \tag{21}$$

Similarly, by (7c) and the fact that $\bar{\pi}_f \geq 0$ we have:

$$\bar{\pi}_f^\top D_f y_f \geq \bar{\pi}_f^\top (b_f - C_f x) \quad \forall f \in \mathcal{F}. \tag{22}$$

From the primal constraint (7e), we have $(b_f - C_f x)^\top \lambda_f \geq d_f^\top y_f$. Since $\bar{\zeta_f} \geq 0$, we have:

$$\bar{\zeta_f} (b_f - C_f x)^\top \lambda_f \geq \bar{\zeta_f} d_f^\top y_f, \quad \forall f \in \mathcal{F}. \tag{23}$$

Left-multiplying the primal equality (7d) by $\bar{\xi}_f^\top$ gives:

$$\bar{\xi}_f^\top D_f \lambda_f = \bar{\xi}_f^\top d_f. \tag{24}$$

Summing (22) over $f = 1, \ldots, n$ gives:

$$\sum_{f=1}^{n} \bar{\pi}_f^\top D_f y_f \geq \sum_{f=1}^{n} \bar{\pi}_f^\top (b_f - C_f x). \tag{25}$$

Next, we add the inequalities (21) and (25):

$$\begin{aligned}
\bar{\mu}^\top (a - Ax) + \sum_{f=1}^{n} \bar{\pi}_f^\top (b_f - C_f x) &\leq \sum_{f=1}^{n} \left(\bar{\mu}^\top B_f + \bar{\pi}_f^\top D_f\right)^\top y_f \\
&= \sum_{f=1}^{n} (\bar{\zeta}_f d_f)^\top y_f && \text{by (14)} \\
&\leq \sum_{f=1}^{n} \bar{\zeta}_f (b_f - C_f x)^\top \lambda_f && \text{by (23)} \\
&\leq \sum_{f=1}^{n} (-D_f^\top \bar{\xi}_f)^\top \lambda_f && \text{by (14)} \\
&= \sum_{f=1}^{n} -\bar{\xi}_f^\top d_f && \text{by (24)}
\end{aligned}$$

Rearranging terms to match the cut form yields

$$\bar{\mu}^\top (a - Ax) + \sum_{f=1}^{n} \left[\bar{\pi}_f^\top (b_f - C_f x) + \bar{\xi}_f^\top d_f\right] \leq 0,$$

which is precisely the cut (17b). Therefore, (17b) is satisfied for all x feasible to (7). □

If the subproblem (12) is infeasible at x^*, then the dual subproblem (13) is unbounded. In practice, the recession direction $\bar{\omega} \in \Omega^R(x^*)$ used to compute the feasibility cut (17b) in Lemma 3.1 is computed by solving the artificially bounded dual subproblem (18) or (19). The following corollary directly connects Lemma 3.1 with the normalized dual subproblems (18) and (19).

Corollary 1. *Let x^* be a candidate solution computed by OBD-MP (17). Then, any feasible solution $\bar{\omega} = (\bar{\mu}, \bar{\pi}, \bar{\xi}, \bar{\zeta})$ to (18) or (19) lies in the recession cone $\Omega^R(x^*)$. Therefore, any inequality of the form (17b) constructed from a feasible solution to (18) or (19) is a valid feasibility cut for (7).*

The result follows directly from the definition of the recession cone $\Omega^R(x^*)$ in (14), since the feasible regions of (18) and (19) are subsets of $\Omega^R(x^*)$. The validity of the resulting feasibility cut then follows from Lemma 3.1. The following example illustrates the construction of feasibility cuts within this framework and demonstrates how the normalization constraint (18d) can lead to tighter cuts.

Example 1. We now consider a simple example to illustrate how feasibility cuts work within this framework. Consider the following bilevel problem:

$$\underset{x \geq 0}{\text{minimize}} \;\; 2x_1 + x_2 \quad \text{s.t.} \quad x_2 \geq x_1, \quad y \in \arg\min\{\bar{y} : \bar{y} \geq 2,\; x_1 + x_2 - \bar{y} \geq 0\} \tag{26}$$

This problem has 1 follower $f = 1$ and is of the form (1) with upper-level data $c_x = (2,1)^\top$ $c_1 = 0, A = \begin{bmatrix}-1 & 1\end{bmatrix}, B = 0, a = 0$ and lower-level data:

$$C_1 = \begin{bmatrix}0 & 0\\ 1 & 1\end{bmatrix}, \quad D_1 = \begin{bmatrix}1\\ -1\end{bmatrix}, \quad b_1 = \begin{bmatrix}2\\ 0\end{bmatrix}, \quad \text{and} \quad d_1 = 1. \tag{27}$$

Using this reformulation, the problem (26) can be reformulated as a single-level problem of the form (7). First, choosing lower bound $\underline{\theta} = 0$, the master problem for the outer Benders problem (17) is

$$\underset{x\geq 0}{\text{minimize}} \quad 2x_1 + x_2 + \theta \quad \text{s.t.} \quad x_2 \geq x_1, \quad \theta \geq 0, \tag{28}$$

with $\mathcal{X} = \{(x_1, x_2) : x_2 \geq x_1\}$ in the context of (17) and no Benders cuts added yet. This feasible region is shown in Figure 4a. Unconstrained in its nonnegative decision x, the leader with chose $x^* = 0$ as its optimal solution at iteration 0. The subproblem (12) is then

$$\underset{y}{\text{minimize}} \quad y \quad \text{s.t} \quad y \geq 2,\ y \leq x_1^* + x_2^*, \tag{29}$$

whose dual is

$$\underset{\pi\geq 0}{\text{maximize}} \quad 2\pi_1 - (x_1^* + x_2^*)\pi_2 \quad \text{s.t.} \quad \pi_1 - \pi_2 \leq 1, \tag{30}$$

where π_1 and π_2 are the multipliers for the constraints $y \geq 2$ and $y \leq x^*$, respectively. Taking $x^* = 0$, the dual problem is unbounded, since π_1 can be made as large as possible with π_2 counteracting in the negative direction. The dual feasible region is $\Omega = \{\pi \geq 0 : \pi_1 - \pi_2 \leq 1\}$ with recession cone $\Omega^R = \{\pi \geq 0 : \pi_1 - \pi_2 \leq 0\}$, shown in Figure 4b, with extreme rays $v^1 = (0,1)^\top$ and $v^2 = (1,1)^\top$.

To provide intuition to considering the recession cone, let (π_1^0, π_2^0) be a feasible point to (30) and (π_1^r, π_2^r) be a direction. For (π_1^r, π_2^r) to be in the recession cone Ω^R, i.e., an unbounded direction, the point $(\pi_1^0 + \alpha\pi_1^r, \pi_2^0 + \alpha\pi_2^r)$ must be feasible for all $\alpha \geq 0$, i.e.,

$$\pi_1^0 + \alpha\pi_1^r - \pi_2^0 + \alpha\pi_2^r \leq 1 \iff \underbrace{\pi_1^0 - \pi_2^0}_{\leq 1} + \alpha(\pi_1^r - \pi_2^r) \leq 1.$$

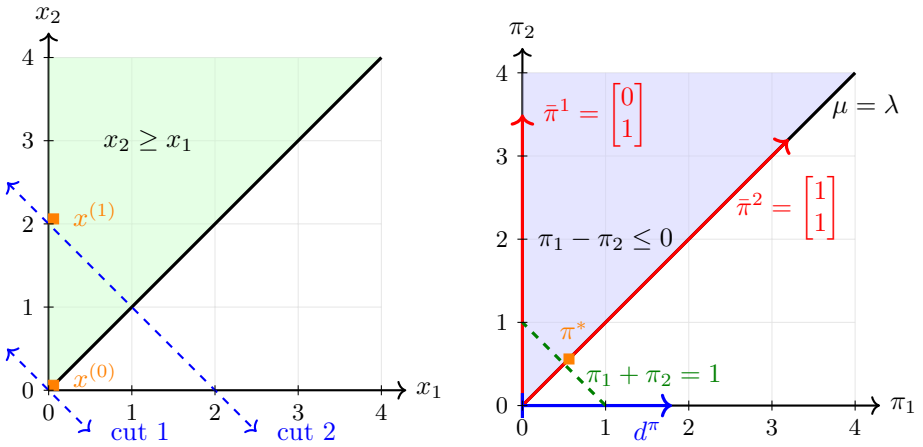
If $\pi_1^r - \pi_2^r > 0$, then as $\alpha \to \infty$ the left-hand-side will eventually become greater than 1 and the constraint will be violated. If $\pi_1^r - \pi_2^r \leq 0$, then $\alpha(\pi_1^r - \pi_2^r) \leq 0$. Since (π_1^0, π_2^0) is a feasible point, $\pi_1^0 - \pi_2^0 \geq 1$, so adding a negative term only makes the left-hand-side smaller, decreasing away from 1 indefinitely as $\alpha \to \infty$. Therefore, a direction $(\pi_1^r - \pi_2^r)$ satisfying $\pi_1^r - \pi_2^r \leq 0$ is unbounded and lies in the recession cone. The feasibility cuts we generate will limit the master problem from exploring such directions.

Since the dual problem is unbounded feasibility cuts are generated and added to the master problem. These cuts have the form (17b):

$$(\bar{\pi}^r)^\top(b_1 - C_1x^*) = \begin{bmatrix}\bar{\pi}_1^r\\ \bar{\pi}_2^r\end{bmatrix}^\top \left(\begin{bmatrix}2\\ 0\end{bmatrix} - \begin{bmatrix}0 & 0\\ 1 & 1\end{bmatrix} x^*\right) \leq 0 \Rightarrow 2\bar{\pi}_1^r - (x_1^* + x_2^*)\bar{\pi}_2^r \leq 0 \tag{31}$$

The question is which extreme ray $\bar{\pi}^r$ to choose for the feasibility cut. For this simple example, it's not hard to see that $\bar{\pi}^2 = (1,1)$ produces the tightest cut $x_1 + x_2 \geq 2$ shown by cut 2 in Figure 4a. However, there are an exponential number of such extreme dual rays in general (Bertsimas and Tsitsiklis 1997), so we use (18) to choose tighter feasibility cut and illustrate this here . The extreme ray subproblem for this example is

$$\underset{\pi\in[0,1]}{\text{maximize}} \quad 2\pi_1 - (x_1^* + x_2^*)\pi_2 \quad \text{s.t.} \quad \pi_1 - \pi_2 \leq 0, \quad \pi_1 + \pi_2 = 1. \tag{32}$$



(A) Primal feasible region of master problem (28).

(B) Recession cone $\Omega^R(x^*)$ of dual feasible region Ω and its extreme rays.

FIGURE 4

Observe that all feasible points to (32) are in the recession cone $\Omega^P R$ shown in Figure 4b which has the optimal solution $\pi^* = (0.5, 0.5)^\top$. From (31), this yields the feasibility cut

$$2(0.5) - (x_1 + x_2)(0.5) \leq 0 \implies x_1 + x_2 \geq 2.$$

Alternatively, choosing the ray $\bar{\pi}^1 = (0, 1)$ yields the feasibility cut $x_1 + x_2 \geq 1$, as shown by cut 1 in Figure 4a, which does not add any additional information to the master problem, since this constraint is redundant to $x_1, x_2 \geq 0$. By selecting the extreme ray in the direction $d^\pi = \pi^\top(b_1 - C_1 x^*) = 2\pi_1$, the extreme ray subproblem (32) yields a tighter feasibility cut.

3.2. **Weighting Problem.** By Corollary 1, any solution to (19) produces a valid feasibility cut. However, as shown in (Seo et al. 2022), including the normalization constraint (18d) yields stronger cuts than solving the problem without this normalization. To preserve the decomposable structure required for the nested BD scheme, we propose a weighting problem that scales the nonzero elements of $\omega = (\mu, \pi, \xi, \zeta)$ computed by the separable problem (19) to produce a vector $\tilde{\omega}$ satisfying the normalization constraint (18d). The motivation for this heuristic is that the recession direction computed by (19) may contain only a few nonzero elements, allowing the weights to be computed cheaply while yielding stronger cuts and thus reducing the number of outer BD iterations.

Denote by $\hat{\omega} = (\hat{\mu}, \hat{\pi}, \hat{\xi}, \hat{\zeta})$ a solution to (19). Our goal now is to compute weights w^μ, w^π, w^ξ and w^ζ on the nonzero elements of $\hat{\omega}$ so that the resulting rays are feasible to (18); i.e., satisfy (18d). Denote by $S^\mu := \{i : \hat{\mu}_i > 0\}$, $S^\pi_f := \{i : \hat{\pi}_{f,i} > 0\}$, $S^\xi_{f,i} := \{i : \hat{\xi}_{f,i} > 0\}$ and $S^\zeta_f := \{i : \hat{\zeta}_{f,i} > 0\}$ the set of nonzero elements of $\hat{\mu}$, $\hat{\pi}$, $\hat{\xi}$ and $\hat{\zeta}$, respectively. Ideally, the number of nonzero elements is much smaller than the original dimension of μ, π and ξ, i.e., $|S^\mu| \ll M_x$, $|S^\pi_f| \ll m_f$ and $|S^\xi_f| \ll n_f$. Let $\hat{\mu}_S$, $\hat{\pi}_{f,S}$, $\hat{\xi}_{f,S}$ and $\hat{\zeta}_{f,S}$ denote vectors only containing the elements of $\hat{\mu}$, $\hat{\pi}_f$, $\hat{\xi}_f$ and $\hat{\zeta}_f$ in S^μ, S^π_f, S^ξ_f and S^ζ_f, respectively. Then we solve the following

weighting problem:

$$\underset{w^\mu, w^\pi, w^\xi, w^\zeta}{\text{maximize}} \quad (w^\mu \circ \hat{\mu}_S)^\top (a - Ax^*) + \sum_{f=1}^{n} (w_f^\pi \circ \hat{\pi}_{f,S})^\top (b_f - Cx^*) + (w_f^\xi \circ \hat{\xi}_{f,S})^\top d_f \tag{33a}$$

$$\text{s.t.} \quad B_f^\top (w^\mu \circ \hat{\mu}_S) + D_f^\top (w_f^\pi \circ \hat{\pi}_{f,S}) - d_f (w_f^\zeta \cdot \hat{\zeta}_{f,S}) = 0 \quad \forall f \in \mathcal{F} \tag{33b}$$

$$D_f (w_f^\xi \circ \hat{\xi}_{f,S}) + (b_f - Cx^*)(w^\zeta \cdot \hat{\zeta}_{f,S}) \leq 0 \quad \forall f \in \mathcal{F} \tag{33c}$$

$$\mathbb{1}^\top (w^\mu \circ \hat{\mu}_S)^\top + \sum_{f=1}^{n} \left[\mathbb{1}^\top (w_f^\pi \circ \hat{\pi}_{f,S}) + \| w_f^\xi \circ \hat{\xi}_{f,S} \|_1 + w_f^\zeta \cdot \hat{\zeta}_{f,S} \right] = 1, \tag{33d}$$

$$w^\mu, w^\pi, w^\xi, w^\zeta \geq 0, \tag{33e}$$

where $\mathbb{1}$ is the vector of ones. The resulting normalized extreme rays are then $\bar{\mu} = w^\mu \circ \hat{\mu}_S$, $\bar{\pi}_f = w_f^\pi \circ \hat{\pi}_{f,S}$, $\bar{\xi}_f = w_f^\xi \circ \hat{\xi}_{f,S}$ and $\bar{\zeta}_f = w_f^\zeta \cdot \hat{\zeta}_{f,S}$ padded with zeros outside of S^μ, S_f^π, S_f^ξ and S_f^ζ, respectively. The resulting vectors are then used to generate feasibility cuts of the form (17b), which are added to the relaxed master problem.

3.3. **Outer Benders Decomposition Algorithm.** The algorithm for the outer BD is presented below in Algorithm 1 and visualized in Figure 3. As discussed in the following section, OBD-XSP (19) and OBD-SP (12) can be solved with an inner Benders decomposition method on lines 5 and 8, respectively.

Algorithm 1 Outer Benders Decomposition for Solving (7)

1: Initialize convergence tolerance $\varepsilon > 0$.
2: **for** $k = 1, 2, \ldots$ **do** ▷ Start Outer Iteration (Benders)
3: Solve OBD-MP (17) to obtain $x^{(k)}$ and $\theta^{(k)}$
4: **if** OBD-SP is unbounded **then**
5: Solve OBD-XSP (19) via Algorithm 3 to obtain extreme rays $\bar{\mu}, \bar{\pi}$ and $\bar{\xi}$
6: Add feasibility cut according to (17b) to OBD-MP
7: **else**
8: Solve OBD-SP (13) via Algorithm 2 to obtain solution μ^*, π^* and ξ^*
9: Compute

$$z^{(k)} = (\mu^*)^\top (a - Ax^{(k)}) + \sum_{f=1}^{n} (\pi_f^*)^\top (b_f - C_f x^{(k)}) + (\xi_f^*)^\top d_f.$$

10: **if** $\| z^{(k)} - \theta^{(k)} \| < \varepsilon$ **then**
11: **break**
12: **else**
13: Add optimality cut according to (17c) to OBP-MP
14: **end if**
15: **end if**
16: **end for**
17: **return** Solution $x^{(k)}$ and dual optimal solution (y^*, λ^*) obtained from OBD-SP (13).

4. Inner Decomposition Algorithm

We apply an inner Benders decomposition (IBD) for the subproblems OBD-SP (13) and OBD-XSP (19). The goal is to increase the efficiency of generating

feasibility and optimality cuts that are passed back to the outer relaxed master problem. The former produces optimal solutions μ^*, π^* and ξ^* to the subproblem (13) that are used to generate optimality cuts of the form (17c), and the latter produces unbounded rays $\bar{\mu}, \bar{\pi}$ and $\bar{\xi}$ that are used to generate feasibility cuts of the form (17b).

4.1. **Inner Decomposition for Generating Optimality Cuts.** Suppose that the outer subproblem OBD-SP (13) is bounded; i.e., x^* is feasible to the primal subproblem (12). Given a candidate solution to the outer subproblem μ^*, the inner subproblem is given by:

$$\begin{aligned}
\text{(IBD-SP)} \quad \underset{\pi,\xi,\zeta}{\text{maximize}} \quad & \sum_{f=1}^{n} \pi_f^\top (b_f - C_f x^*) + \xi_f^\top d_f && && \text{(34a)}\\
\text{s.t.} \quad & D_f^\top \pi_f - \zeta_f d_f = c_f - B_f^\top \mu^* && (\gamma_f) \quad \forall f \in \mathcal{F}, && \text{(34b)}\\
& D_f \xi_f + (b_f - C_f x^*)\zeta_f \leq 0 && (\psi_f) \quad \forall f \in \mathcal{F}, && \text{(34c)}\\
& \pi, \zeta \geq 0, \ \xi \text{ free.} && && \text{(34d)}
\end{aligned}$$

For a fixed candidate solution x^*, define $\tilde{\Omega}^E(x^*)$ to be the set of extreme points of the feasible region of (34) and $\tilde{\Omega}^R(x^*)$ the set of extreme rays of the recession cone to the dual feasible region

$$\tilde{\Omega}^R(x^*) = \left\{ \begin{array}{ll} (\gamma, \psi) : D_f \gamma_f \geq b_f - C_f x^*, \ D_f^\top \psi_f = d_f & \forall f \in \mathcal{F} \\ (b_f - C_f x^*)^\top \psi_f - d_f^\top \gamma_f \geq 0 & \forall f \in \mathcal{F} \\ \psi_f \geq 0, \ \gamma_f \text{ free} & \forall f \in \mathcal{F} \end{array} \right\}. \tag{35}$$

Then the master problem for the inner Benders decomposition to solve (13) is

$$\begin{aligned}
\text{(IBD-MP)} \quad \underset{\mu,\eta}{\text{maximize}} \quad & \mu^\top (a - A x^*) + \eta && && \text{(36a)}\\
\text{s.t.} \quad & \sum_{f=1}^{n} (\bar{\gamma}_f^r)^\top (c_f - B_f^\top \mu) \leq 0 && \forall \bar{\gamma}_f^r \in \tilde{\Omega}^R(x^*) && \text{(36b)}\\
& \sum_{f=1}^{n} (\gamma_f^e)^\top (c_f - B_f^\top \mu) \leq \eta && \forall \gamma_f^e \in \tilde{\Omega}^E(x^*) && \text{(36c)}\\
& \mu \geq 0 && && \text{(36d)}
\end{aligned}$$

where (36b) and (36c) represent the feasibility and optimality cuts of the inner BD loop, respectively. Here, η is an auxiliary variable that approximates the optimal value of the subproblem (34). Similar to ζ_f in the context of the outer subproblem (12), the Lagrange multiplier ψ_f is associated with the primal constraint (34c) which has a zero right-hand side. Consequently, ψ_f does not appear in the cut expressions (36b) or (36c). Notably, the subproblem (34) is fully separable for follower f, so that (34) can be solved in parallel. The complete inner decomposition scheme for solving (13) is provided in Algorithm 2.

4.2. **Inner Decomposition for Generating Feasibility Cuts.** Now, consider the relaxed outer extreme ray subproblem OBD-XSP (19). The subproblem solved

[1]The inner iteration counter ν is indexed by the outer iteration k, i.e., $\nu \equiv \nu_k$, denoting the νth inner iteration associated with outer iteration k. However, we omit the subscript k for notational brevity.

Algorithm 2 Inner Benders Decomposition for Solving (13)

Input: Outer iteration counter k and candidate OBD solution $x^{(k)}$.[1]
for $\nu = 1, 2, \ldots$ **do** ▷ Start Inner Iteration
  Master Problem Solve. Solve IBD-MP (36) to obtain $\mu^{(\nu)}$ and $\eta^{(\nu)}$.
  Subproblem Solve. Solve IBD-SP (34).
  if IBD-SP is unbounded **then**
    Retrieve unbounded dual rays $\bar{\gamma}^r$.
    Add feasibility cut according to (36b) to IBD-MP
  else
    Retrieve optimal solution π^* and ξ^*.
    Compute

$$z^{(\nu)} = \sum_{f=1}^{n} (\pi_f^*)^\top (b_f - C_f x^{(k)}).$$

    if $\|z^{(\nu)} - \eta^{(\nu)}\| < \varepsilon$ **then**
      break
    else
      Add optimality cut according to (36c) to IBD-MP
      $\nu \leftarrow \nu + 1$
    end if
  end if
end for
return Solution $\mu^{(\nu)}, \pi^*$ and ξ^*

at each inner iteration is

$$\text{(IBD-XSP)} \quad \underset{\pi,\zeta,\xi}{\text{maximize}} \quad \sum_{f=1}^{n} \pi_f^\top (b_f - C_f x^*) \tag{37a}$$

$$\text{s.t.} \quad D_f^\top \pi_f - \zeta_f d_f = -B_f^\top \mu^* \qquad (\gamma_f) \quad \forall f \in \mathcal{F} \tag{37b}$$

$$D_f \xi_f + (b_f - C_f x^*)\zeta_f \leq 0 \qquad (\psi_f) \quad \forall f \in \mathcal{F}, \tag{37c}$$

$$\pi, \zeta \in [0, 1], \ \xi \in [-1, 1] \tag{37d}$$

which is again fully separable for follower f. The master problem for the inner Benders decomposition when solving (18) is the same as (36). An additional, optional step in the inner scheme, when generating feasibility cuts for the outer problem, is to apply the weighting method described in Section 3.2. The complete inner scheme for generating feasibility cuts for OBD-MP is presented in Algorithm 3.

5. Parallel Implementation

The nested BD framework is implemented on a distributed-memory architecture to enable scalable parallel execution. The algorithm is developed in C++ using a hybrid parallelization strategy: the Message Passing Interface (MPI) is employed for inter-processor communication across distributed memory, while Open Multi-Processing (OpenMP) is used for shared-memory parallelization within each processor, particularly for computationally intensive linear algebra operations such as matrix multiplication in model formulations and subproblem updates. This section describes important aspects of this implementation, particularly subproblem partitioning and MPI communication.

5.1. **Subproblem Partitioning.** The key advantage of the proposed approach is that the inner subproblem (34) or (37) fully decouples by follower. Consequently, the follower set can be partitioned and the corresponding subproblems distributed

Algorithm 3 Inner Benders Decomposition for Solving (18)

Input: Outer iteration counter k and candidate OBD solution $x^{(k)}$.[2]
for $\nu = 1, 2, \ldots$ **do** ▷ Start Inner Iteration
  Master Problem Solve. Solve IBD-MP (36) to obtain $\mu^{(\nu)}$ and $\eta^{(\nu)}$
  Subproblem Solve. Solve IBD-XSP (37).
  if IBD-SP is unbounded **then**
    Retrieve unbounded dual rays $\bar{\gamma}^r$.
    Add feasibility cut according to (36b) to IBD-MP
  else
    Retrieve optimal solution π^*.
    Compute

$$z^{(\nu)} = \sum_{f=1}^{n} (\pi_f^*)^\top (b_f - C_f x^{(k)}).$$

    if $\|z^{(\nu)} - \eta^{(\nu)}\| < \varepsilon$ **then**
      break
    else
      Add optimality cut according to (36c) to IBD-MP
    end if
  end if
end for
(Optional) Compute normalization weights by solving (33) to obtain $\bar{\mu}, \bar{\pi}$ and $\bar{\xi}$
return Unbounded rays $\bar{\mu}, \bar{\pi}$ and $\bar{\xi}$

across worker processors. Let p denote the number of processors (i.e., tasks), and let $\mathcal{F}_i \subseteq \mathcal{F}$ represent the subset of followers assigned to processor $i \in \{1, \ldots, p\}$ such that that $\mathcal{F} = \bigcup_{i=1}^{p} \mathcal{F}_i$ and $\mathcal{F}_i \cap \mathcal{F}_j = \emptyset$ for $i \neq j$. The inner subproblem solved by processor i when generating feasibility cuts for the outer master problem, denoted IBD-XSP$_i$, is

$$\text{(IBD-XSP}_i\text{)} \quad \underset{\pi}{\text{maximize}} \quad \sum_{f \in \mathcal{F}_i} \pi_f^\top (b_f - C_f x^*) \tag{38a}$$

$$\text{s.t.} \quad D_f^\top \pi_f - \zeta_f d_f = -B_f^\top \mu^* \qquad (\gamma_f) \quad \forall f \in \mathcal{F}_i \tag{38b}$$

$$D_f \xi_f + \zeta_f (b_f - C_f x^*) \leq 0 \qquad (\psi_f) \quad \forall f \in \mathcal{F}_i \tag{38c}$$

$$\pi_f, \zeta_f \in [0, 1], \ \xi_f \in [-1, 1] \qquad \forall f \in \mathcal{F}_i, \tag{38d}$$

The corresponding optimality-cut subproblem, denoted IBD-SP$_i$, is defined analogously with respect to (34). Upon completion, each processor returns its partial objective contribution, associated solution (π_f, ξ_f, ζ_f) and dual multiplier γ_f for all $f \in \mathcal{F}_i$ to the master process. The master process then aggregates the results from each processor to construct the corresponding Benders cut.

While the partitioning of followers does not impact the validity of the algorithm, it can significantly affect overall execution time. In particular, the next inner BD iteration cannot begin until all worker processors have completed their assigned subproblem, making load balance (i.e., the even distribution of computational work across processors) critical to performance. In applications where follower subproblems differ structurally (e.g., when some followers involve substantially more decision variables than others) careful workload distribution is essential to avoid idle processor time. If the number of followers equals the number of available processors (i.e., $|\mathcal{F}| = p$), one could assign a single follower to each processor. However, this strategy may introduce substantial communication and coordination overhead, and for instances with thousands of followers it may be infeasible given

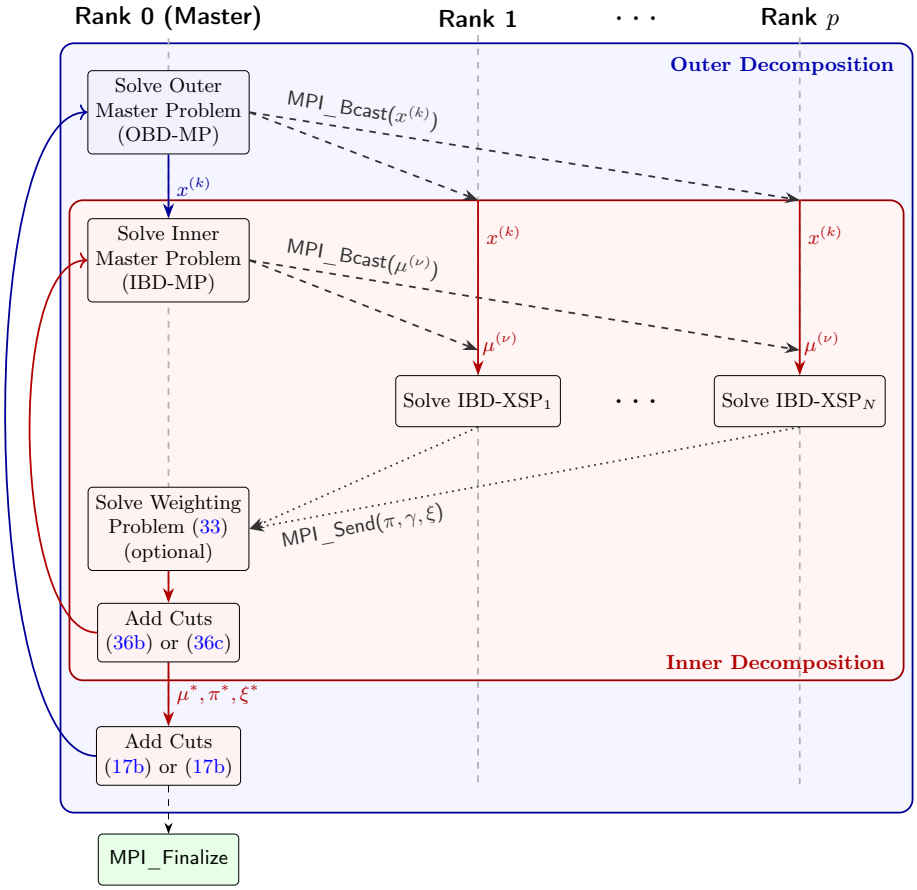


FIGURE 5. Distributed implementation of the nested Benders decomposition framework using MPI. Dashed arrows represent MPI broadcast communications, and dotted arrows represent blocking point-to-point MPI calls.

available computing resources. In the application described in Section 6.1, follower subproblems are structurally similar, and we therefore use a uniform distribution of followers among processors; i.e., $|\mathcal{F}_i| = \lfloor |\mathcal{F}|/p \rfloor$ for $i \in \{1, \ldots, p-1\}$, with the remaining followers assigned to process p.

5.2. **MPI Communication.** The coordination and communication between processors is a crucial aspect of the parallel implementation of the nested BD scheme. Processors are identified by their rank, with rank 0 representing the master processor and ranks 1 through p representing the workers. The master processor (rank 0) solves the outer and inner BD master problems, and constructs feasibility and optimality cuts, while the workers solve the inner subproblem (38) for their assigned partition of followers. Consequently, the master processor performs the majority of the computational work, while the worker processors remain idle until they are needed to solve the subproblem. Because the subproblem requires information from the master problem to formulate its objective and constraints – specifically $x^{(k)}$ and $\mu^{(\nu)}$ – these data must be communicated to each worker process at every inner iteration.

Figure 5 shows the sequence of solves and MPI communications at outer iteration k and inner iteration ν of the nested BD framework. The outer iteration k starts

with the solve of OBD-MP (17) on the master process. Upon completion, the master process broadcasts the candidate upper-level solution $x^{(k)}$ to each of the worker processes. Since the candidate solution $x^{(k)}$ remains constant across all nested inner iterations for a fixed outer iteration k, this candidate solution only needs to be communicated to the worker processors once per outer iteration. Then, the master processor begins the inner decomposition by solving IBD-MP (36). The candidate solution $\mu^{(\nu)}$ is then broadcast to each of the worker threads for their parallel solve of the partitioned subproblem IBD-SP (34) or IBD-XSP (37) (depending on whether optimality or feasibility cuts need to be generated).

Once each process i solves their respective subproblem, it sends the components (π_f, ξ_f) and γ_f of the primal and dual solutions for all $f \in \mathcal{F}_i$, respectively, to the master processor via a blocking point-to-point communication,[3] highlighted by the dotted arrows in Figure 5. This blocking call is met with a matching receive call from the master processor (not shown in Figure 5). The master processor then aggregates the solutions received from all worker processes. Crucially, the master process must wait until all workers have completed their subproblem solves. Therefore, effective load balancing is essential to ensure that no processor becomes a bottleneck. If the convergence criteria is met for the inner scheme, then the components μ^*, π^* and ξ^* of the solution to (12) are returned from the inner iteration to construct a cut for the outer master problem. If not, the unbounded dual ray is optionally weighted by solving problem (33), after which a feasibility or optimality cut is generated via (36c) or (36b) by the master processor for the inner master problem.

5.3. **Benders Enhancements.** To further accelerate the nested Benders scheme, we incorporate advanced cut generation techniques for the master problem from the literature. These methods aim to further reduce the number of outer Benders iterations required for convergence and to speed up master problem solves. Accordingly, they are integrated within the parallel framework described in the previous section.

We implement disaggregated multi-cut generation for optimality cuts, originally introduced in the context of the L-shaped method for stochastic recourse problems; see, e.g., Birge and Louveaux (1988) and Laporte and Louveaux (1993). This approach has since been extended to more general bundled cut generation schemes, including covering cut bundles (Saharidis et al. 2010), and more recently in (Kaltis and Saharidis 2026). The key observation is that individual optimality cuts often involve only a small subset of MP decision variables and therefore have relatively low-density, limiting their impact on the feasible region of the MP. By contrast, aggregating these cuts yields higher-density cuts that impose stronger restrictions on the MP. At the same time, disaggregation allows components of the optimal value function to tighten independently, which further accelerates convergence by improving the approximation of the value function across iterations.

To apply this within our framework, we introduce the auxiliary variable θ_f for each follower f, which represents the optimal value of follower f's objective in the MP. The OBD-MP objective function (17a) is then modified:

$$c_x^\top x + \sum_{f=1}^{n} \theta_f. \tag{39}$$

[3]A blocking point-to-point communication is a message-passing operation between two processors in which the sending (or receiving) process does not proceed until the communication has been completed, i.e., the message has been fully transmitted (or received) and the corresponding buffer can be safely reused. In MPI, this typically corresponds to calls such as `MPI_Send` and `MPI_Recv` (Hager and Wellein 2010).

At each outer iteration, rather than adding a single optimality cut of the form (17c), we add $|\mathcal{F}|$ cuts of the form:

$$\frac{1}{|\mathcal{F}|}\left[(\mu^e)^\top(a - Ax)\right] + (\pi_f^e)^\top(b_f - C_f x) + (\xi_f^e)^\top d_f \leq \theta_f. \tag{40}$$

These cuts represent a disaggregated version of the single cut in that the sum of (40) across followers yields (17c). However, when enforced together, these cuts are collectively more restrictive than the aggregated cut (17c). When $\mu = 0$, these cuts are an exact disaggregation. The benefit of this approach is that each θ_f tightens independently, rather than allowing improvements for one follower to compensate for weaker cuts associated with another in (17c).

A potential drawback of the disaggregated multi-cut routine is that up to $|\mathcal{F}|$ cuts may be added per iteration, compared to one in the single-cut approach. This can cause the master problem to grow significantly over many iterations. To address this, we selectively add optimality cuts (40) using a violated-cut criterion (Holmberg 1990). Let $g_f(x^k)$ denote the left-hand side of the optimality cut (40) evaluated at the current candidate solution x^k, and let θ_f^k denote the current value of the corresponding value function approximation, i.e., the optimal value of θ_f produced by OBD-MP (17) with objective (39) at iteration k. The optimality cut for follower f is added to OBD-MP only if it is violated by the current solution, i.e., $g_f(x^k) > \theta_f^k + \varepsilon$, where $\varepsilon > 0$ is a small numerical tolerance, taken to be $\varepsilon = 10^{-6}$ for this implementation. Cuts that are already satisfied by x^k are not added, thus avoiding polluting the master with redundant constraints that do not tighten the current relaxation. While this approach limits master problem growth without sacrificing convergence, more advanced size management techniques exist in the literature, such as age-based pruning and slack analysis; see, e.g., Pacqueau et al. (2012) and Papadakos (2008). These techniques are not considered in this work, but represent an area for future research.

6. Numerical Experiments

We now outline numerical experiments demonstrating the practical application of the proposed nested BD framework. Specifically, we apply the method to a public transportation planning problem to evaluate the parallel performance of the algorithm implementation and compare with a monolithic baseline. We begin by describing the line-planning problem with integrated passenger routing, then outline the experimental design and present the results of the parallel algorithm.

6.1. Line Planning with Integrated Passenger Routing.

We apply the proposed nested BD scheme to the line-planning problem with passenger routing problem from (Schöbel and Scholl 2006). Surveys on the topic can also be found in (Schmidt and Schöbel 2024; Schöbel 2012). Let $\mathcal{N} = (S, E)$ be a public transportation network (PTN) represented as a undirected graph with node set S representing stations and edge set E, where each edge $\{u, v\}$ represents a direct connection from station u to v, e.g. a track or street. Let $\mathcal{L}$ be a line pool consisting of a set of paths in the PTN. Denote by $E(\ell)$ the set of edges belonging to line $\ell \in \mathcal{L}$ and $\mathcal{L}(u) = \{\ell \in \mathcal{L} : u \in \ell\}$ as the set of all lines passing through station u. The line-planning problem is then to choose a subset of lines $L \subseteq \mathcal{L}$ that allows each customer to travel from their origin to destination and balances two objectives: (1) the cost to the public transportation planner, and (2) the inconvenience of the customers, as measured by accumulated flow costs.

In the PTN $\mathcal{N}$, we denote the set of all origin-destination pairs (s, t) by the set $\mathcal{R} \subset S \times S$. As conventional in line-planning models, we construct a directed graph

from $\mathcal{N}$ to model the problem, referred to as the change & go network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and constructed as follows. The set of vertices $\mathcal{V}$ is the union of the following sets:

- $\mathcal{V}_{CG} := \{(s, \ell) \in S \times \mathcal{L} : \ell \in \mathcal{L}(s)\}$ (set of all station-line pairs)
- $\mathcal{V}_S := \{(s, 0) : s \in S\}$ (set of stations)

And the set of edges $\mathcal{E}$ is the union of the following sets:

- $\mathcal{E}_{go} := \bigcup_{\ell \in \mathcal{L}} \mathcal{E}^\ell$ (driving edges)
- $\mathcal{E}_{BA} := \{((s, 0), (s, \ell)) \in \mathcal{V}_S \times \mathcal{V}_{CG}$ and $((s, \ell), (s, 0)) \in \mathcal{V}_{CG} \times \mathcal{V}_S : s \in \ell, s \in S\}$ (boarding/alighting edges),

where

$$\mathcal{E}^\ell := \{((s, \ell), (s', \ell)) \in \mathcal{V}_{CG} \times \mathcal{V}_{CG} : (s, s') \in \ell\}$$

is the set of the driving edges of line $\ell \in \mathcal{L}$. We can now define the edge costs d_e to represent the inconvenience cost for a passenger traveling on edge $e \in \mathcal{E}$:

$$d_e = \begin{cases} (\text{change time})/2 & \text{if } e \in \mathcal{E}_{BA} \\ \text{travel time} & \text{if } e \in \mathcal{E}_{go}. \end{cases}$$

Note that the change time is divided equally across the corresponding transfer edges to avoid double counting in the path cost.

The line-planning problem with minimal travel times (LPMT), as presented in (Schöbel and Scholl 2006) is then

$$\begin{aligned}
\underset{x,y}{\text{minimize}} \quad & \sum_{r \in \mathcal{R}} \rho_r \sum_{e \in \mathcal{E}} d_e y_r^e && && (41a) \\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} \sum_{e \in \mathcal{E}^\ell} y_r^e \le |\mathcal{R}||\mathcal{E}^\ell| x_\ell && \forall \ell \in \mathcal{L} && (41b) \\
& \Theta y_r = b_r && \forall r \in \mathcal{R} && (41c) \\
& \sum_{\ell \in \mathcal{L}} c_\ell x_\ell \le \Delta && && (41d) \\
& y_r^e, x_\ell \in \{0, 1\} && \forall r \in \mathcal{R}, e \in \mathcal{E}, \ell \in \mathcal{L}, && (41e)
\end{aligned}$$

where

- $\mathcal{R} \subseteq S \times S$ is the set of OD pairs, expressed as tuples $r = (s, t)$;
- d_e is the cost of traveling on edge $e \in \mathcal{E}$;
- $\Theta \in \mathbb{Z}^{|\mathcal{V}| \times |\mathcal{E}|}$ is the node-arc-incidence matrix of the change & go network $\mathcal{G}$;
- $b_r \in \mathbb{R}^{|\mathcal{V}|}$ is the demand vector;
- ρ_r is the number of customers that wish to travel from origin s to destination t; i.e., OD pair $r = (s, t)$
- c_ℓ is the cost of opening line $\ell \in \mathcal{L}$;
- Δ is the line planner's budget;
- $y_r^e \in \{0, 1\}$ indicates whether edge e is used in a shortest path from origin s to destination t for OD pair $r = (s, t)$; and,
- $x_\ell \in \{0, 1\}$ indicates whether line ℓ is opened.

Note that the incidence matrix Θ is identical for each OD pair r, but the right-hand-side b_r is specific to an OD pair r. Moreover, alternative formulations of the LPMT (41) exist in which line frequencies are explicitly incorporated. In these settings, x_ℓ is restricted to the natural numbers rather than binary. We do not consider the frequency-based formulation in this paper.

6.1.1. *Line-Planning Cost Formulation.* The formulation (41) is referred to as the "customer-oriented" approach, which minimizes the *total* inconvenience incurred by all passengers. Alternatively, we could formulate (41) as a bilevel optimization problem, with the transportation planner as the upper-level player (leader) and

each OD pair $r = (s,t) \in \mathcal{R}$ as the followers. The bilevel line-planning problem (LPMT-BL) is then

$$\underset{x,y}{\text{minimize}} \quad \sum_{\ell \in \mathcal{L}} c_\ell x_\ell \tag{42a}$$

$$\text{s.t.} \quad \sum_{r \in \mathcal{R}} \sum_{e \in \mathcal{E}^\ell} y_r^e \le |\mathcal{R}||\mathcal{E}^\ell| x_\ell \qquad \forall \ell \in \mathcal{L} \tag{42b}$$

$$y_r \in \underset{\bar{y}_r \in [0,1]}{\arg\min} \left\{ \sum_{e \in \mathcal{E}} d_e \bar{y}_r^e : \Theta \bar{y}_r = b_r, y_r^e \le x_\ell \ \forall \ell \in \mathcal{L}, e \in \mathcal{E}^\ell \right\} \quad \forall r \in \mathcal{R} \tag{42c}$$

$$x_\ell \in \{0,1\} \qquad \forall \ell \in \mathcal{L} \tag{42d}$$

Now, the leader's objective is to minimize the total cost of opening up all lines. Then, the lower-level problem is to minimize the total inconvenience of the route for each OD pair subject to shortest path constraints. Note that we have relaxed the integrality constraint on the lower-level variables $y_r^e \in [0,1]$ so that they represent the fraction of customers traveling on OD pair $r = (s,t)$ using edge e. As a result, strong duality holds for the lower-level problem (42c). The decomposable structure and strong-duality formulation of the bilevel problem (42) is outlined in Appendix A.

6.1.2. *Total Travel Time Formulation.* We propose an alternative formulation that minimizes weighted passenger travel time at the upper level rather than cost. In this formulation, the line-planning cost is implemented as an ε-constraint:

$$\underset{x,z,y}{\text{minimize}} \quad \sum_{r \in \mathcal{R}} \rho_r \sum_{e \in \mathcal{E}} d_e y_r^e \tag{43a}$$

$$\text{s.t.} \quad \sum_{\ell \in \mathcal{L}} c_\ell x_\ell \le \Delta \tag{43b}$$

$$(42b), (42c) \tag{43c}$$

$$x_\ell \in \{0,1\} \ \forall \ell \in \mathcal{L}. \tag{43d}$$

This formulation is similar to the single-level LPMT formulation (41), with the addition of the lower-level problem (42c). The motivation for this formulation stems from the observation that the line-planning cost model (42) grants substantial control to the line planner (upper-level decision-maker). Our initial experiments show that its solutions are not passenger-friendly: the planner selects the cheapest set of lines that ensures feasibility, without regard for realized passenger travel times.

In contrast, the total travel time formulation (43) is more passenger-centric, as it seeks to minimize weighted travel time subject to a line-planning budget Δ. This contrast is illustrated in the solutions from each formulation on the Ring 3x3 and 4x4 datasets shown in Appendix B. From a experimental perspective, this formulation is particularly valuable for gaining insights to the nested Benders framework because follower (subproblem) variables enter the upper-level objective, necessitating optimality cuts to approximate the value function of the lower-level problems. While (42) permits analysis of feasibility cut generation, (43) enables the study of optimality cut generation within the proposed approach.

6.1.3. *Master Problem Cuts.* To solve the bilevel line-planning formulations with the nested Benders approach, we initialize the master problem with problem-specific constraints to accelerate convergence. Specifically, we add the constraint

$$\sum_{v \in \mathcal{V}} \sum_{\ell \in \mathcal{L}(v)} x_\ell \ge 1, \tag{44}$$

which states that every node (i.e., stop) must have at least one line going through it. While this constraint is likely loose, it aids the master problem in computing feasible solutions early on.

In the context of the disaggregated multi-cut generation for optimality cuts described in Section 5.3, we also add a lower-bound on the value of passenger r's objective θ_r in OBD-MP (17). These cuts are only generated for the total travel time formulation (43), where follower (subproblem) variables appear in the master problem objective. The bound $\underline{\theta}_r$ represent a lower bound on the minimum travel time through the network for passenger r. These bounds are computed by solving the lower-level problem (42c) with all lines open (i.e., $x_\ell = 1$ for all $\ell \in \mathcal{L}$). Let $\hat{y}_r$ denote an optimal solution to the LP (42c) with $x_\ell = 1$ for all $\ell \in \mathcal{L}$, then the lower bound on θ_r is

$$\underline{\theta}_r = \rho_r \sum_{e \in \mathcal{E}} d_e \hat{y}_r^e. \tag{45}$$

The constraint $\underline{\theta}_r \leq \theta_r$ for all $r \in \mathcal{R}$ is then included in the master problem.

6.2. **Strong Scaling Analysis.** We conduct numerical experiments using the parallel implementation of the nested BD scheme described in Section 5. The goal of these experiments is to analyze how computational time is distributed across components of the algorithm and how this breakdown changes as the number of processors increases. Therefore, we use *strong* scaling analysis, increasing the number of processors while maintaining a fixed problem size for each instance. This contrasts with *weak* scaling, in which both the number of processors and the problem size are increased proportionally.[4]

6.2.1. *Experimental Design.* We apply the nested BD scheme to 10 instances of the LPMT cost formulation (42) with varying sizes. The datasets are obtained from LinTim, a scientific software toolbox for public transportation planning (Schiewe et al. 2025, 2026). The datasets are summarized in Table 2 and vary in size based on the the PTN $\mathcal{N}$, the line pool $\mathcal{L}$ and set of OD pairs (i.e., followers) $\mathcal{R}$. The line pool and change & go network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ are generated using the LinTim software. According to the bilevel LPMT (42) problem description in Section 6.1, the number of lower-level variables and constraints for OD pair (i.e., follower) f is $n_f = |\mathcal{E}|$ and $m_f = |\mathcal{V}| + \sum_{\ell \in \mathcal{L}} |\mathcal{E}^\ell|$, respectively, for $f = 1, \ldots, n$. Therefore, the number of lower-level variables in the whole system is $N_y = n \cdot n_f$, and the number of lower-level constraints in the whole system is $M_y = n \cdot m_f$. Since the upper-level player in the bilevel LPMT (42) decides which lines to open, the number of upper-level decisions N_x is equal to the number of lines $|\mathcal{L}|$.

We test the nested BD scheme on the 10 LPMT problem instances described in Table 2 and deploy the parallel code on a high-performance computing (HPC) cluster. Compute nodes on the cluster are equipped with 128 cores and 512 GB of memory per node (4 GB per core). Each node runs on AMD EPYC 7763 (Zen 3) processors, with 64 cores per CPU and a base clock speed of 2.45 GHz. Each problem instance is executed using $p \in \{1, 2, 4, 8, 16, 32, 64, 128\}$ MPI processes on a single compute node containing 128 cores. For each configuration, the total number of cores per node is fixed at 128, and OpenMP threads are assigned dynamically so that each MPI process uses $128/p$ threads. For example, when $p = 16$, each MPI process is allocated 8 OpenMP threads.

[4]For example, in strong scaling we solve a problem instance with 10^6 variables while increasing the number of processors from 1 to 128 to measure runtime reduction. In contrast, under weak scaling, the problem size would be increased proportionally with the number of processors, e.g., from 10^4 variables on 1 processor to 1.28×10^6 variables on 128 processors, so that the workload per processor remains approximately constant.

Dataset	OD Pairs $\lvert\mathcal{R}\rvert$	Lines $\lvert\mathcal{L}\rvert$	Nodes $\lvert\mathcal{V}\rvert$	Edges $\lvert\mathcal{E}\rvert$	LL Vars. N_y	LL Cons. M_y
Mandl	172	19	107	330	56,760	175,440
Sioux Falls	552	50	264	860	474,720	1,450,656
Erding	675	66	444	1,436	969,300	2,978,100
Athens	2,385	59	741	2,642	6,301,170	19,146,780
Bahn	6,106	132	2,418	8,408	51,339,248	157,070,744
Ring 3×3	90	81	272	886	79,740	241,020
Ring 4×4	272	254	983	3,356	912,832	2,747,744
Ring 5×5	650	609	2,627	9,186	5,970,900	17,946,500
Ring 6×6	1,332	1,040	4,916	17,436	23,224,752	69,772,824
Ring 7×6	1,806	1,341	6,615	23,606	42,632,436	128,052,624

Table 2. Summary of datasets (LL = lower level).

The framework is implemented in C++17 using OpenMP (via GCC 11.3.0) for multi-threaded matrix operations and OpenMPI 4.1.5 for communication between master and worker processes. All optimization problems are solved using Gurobi 13.0.0. The full codebase and data are publicly available to ensure reproducibility.[5]

6.2.2. *Parallel Computing Metrics & Performance Profiling.* The subsequent analysis employs standard HPC metrics that are not commonly used in the operations research literature. For completeness, these metrics are briefly defined below. A comprehensive treatment of HPC terminology and performance analysis can be found in (Hager and Wellein 2010).

For each run, the wall time and CPU time are recorded to construct a comprehensive performance profile. Wall time (i.e., elapsed or clock time) measures the total real-world duration from the start to the completion of a task. In contrast, CPU time captures the cumulative time that processors actively spend executing instructions for the task. In a parallel setting, wall time captures the effective runtime experienced by the user, while CPU time may exceed wall time due to concurrent execution across multiple processors. Let $T(p)$ denote the wall time on p processors. For a perfectly parallel task with no idle time, the total CPU time is approximately $p \cdot T(p)$.

We utilize several metrics to quantify the performance of the parallel implementation. The first is the speedup on p processors, defined as $S(p) = T(1)/T(p)$. A theoretical upper bound on achievable speedup is given by Amdahl's law, which states that if a fraction α of the computation is serial, then $S(p) \leq 1/(\alpha + (1-\alpha)/p)$ (Amdahl 1967). As $p \to \infty$, the speedup is bounded by $1/\alpha$, highlighting the impact of serial bottlenecks. In the parallel nested BD scheme (see Figure 5), such bottlenecks include tasks executed on the master process, e.g., master problem solves and Benders cut generation.

To assess resource utilization, we compute the CPU efficiency $E_{\text{CPU}}(p) = T_{\text{CPU}}(p)/(p \cdot T(p))$, where $T_{\text{CPU}}(p)$ denotes the total aggregated CPU time across processors. This metric measures how effectively computational resources are used, with $E_{\text{CPU}}(p) = 100\%$ indicating perfect utilization.

6.3. **Results for Line Cost Formulation.** We first apply the nested Benders scheme to the line-planning cost formulation (42) described in Section 6.1.1. Later on, in Section 6.4, we analyze the results on the alternative total travel time formulation (43) described in Section 6.1.2. For the following results, we do not use the normalization weighting scheme presented in Section 3.2. In the last

[5] https://github.com/dominicflocco/bilevel-decomp

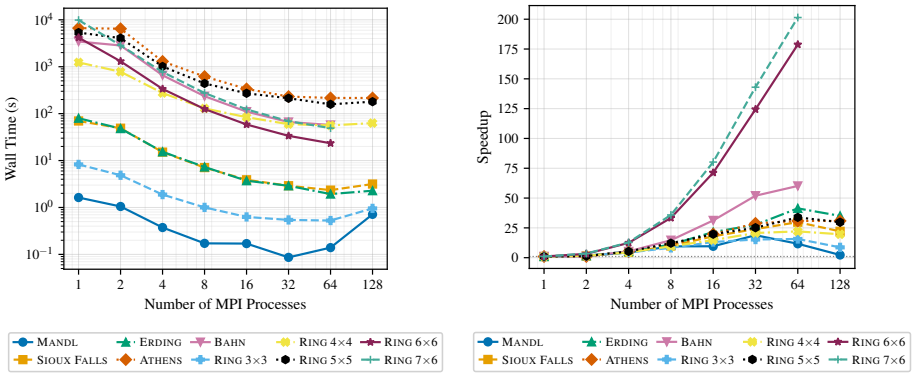


FIGURE 6. Performance of parallel nested Benders. (Left) Walltime (s) by number of MPI processes. (Right) Speedup $S(p) = T(1)/T(p)$

Subsection 6.3.4, we compare results with and without the normalization scheme applied.

6.3.1. *Strong Scaling Analysis.* Figure 6 presents two performance profiles of the parallel nested Benders scheme on 10 line-planning instances using the line cost formulation (42). The left plot reports wall-clock time as a function of the number of MPI processes. As expected, wall time generally decreases as the number of processes increases, demonstrating that the parallel scheme effectively leverages additional computational resources.

However, for smaller datasets, namely Mandl and Ring 3x3, performance begins to degrade or plateau at higher process counts (e.g., 128 processes). This behavior is likely driven by communication overhead: the latency associated with message passing eventually outweighs the computational gains from additional processors. This effect is particularly pronounced for the smaller instances, Mandl and Ring 3x3, where the per-process workload is insufficient to offset communication costs.

For the three largest datasets – Ring 6x6, Ring 7x6, and Bahn – results with 128 MPI processes are not reported. This limitation arises from memory constraints of the computing architecture, which provides 4 GB of RAM per core. When assigning one process per core, the available memory is insufficient to store the data required for the follower subproblems in these larger instances. Consequently, experiments with 128 MPI processes were not conducted for these instances.

The right plot in Figure 6 shows the speedup $T(p)/T(1)$ as a function of the number of MPI processors p. A similar trend is observed: speedup generally increases with the number of processes. Ideal strong scaling corresponds to linear speedup (i.e., $T(p)/T(1) = p$), which is closely approached for many datasets up to 64 processes. Consistent with the wall-time results, the speedup for smaller datasets plateaus or even declines at 128 processes, reflecting the growing impact of communication latency.

Notably, the largest datasets – Ring 6x6, Ring 7x6, and Bahn – appear as clear outliers, exhibiting superlinear speedup and thus seemingly exceptional strong scaling. However, a closer inspection of the performance data in Table 3 reveals that these instances converge in a single Benders iteration. This eliminates much of the algorithm's inherent serial overhead, leading to artificially inflated speedups. In effect, only one feasibility cut is required to solve the bilevel problem for these instances. In Section 6.4, we examine an alternative formulation for which these

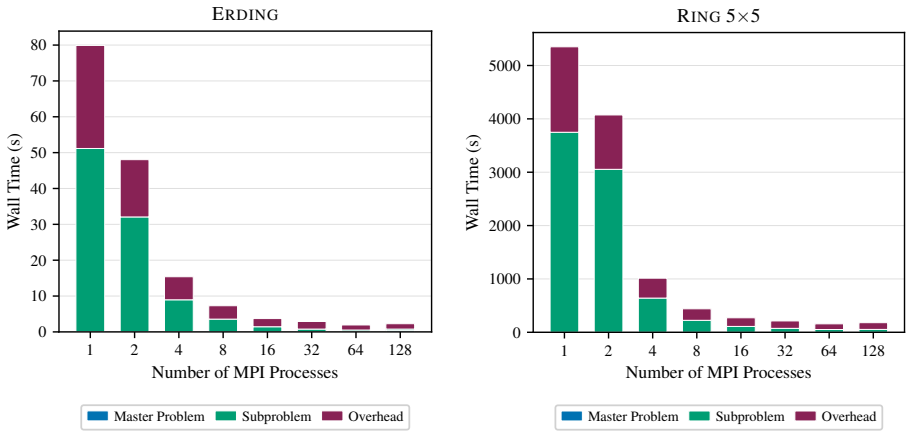


FIGURE 7. Walltime (s) breakdown for (Left) Erding and (right) Ring 5x5 datasets.

datasets require multiple iterations, yielding a more representative assessment of parallel performance.

6.3.2. *Time Profiling.* Figure 7 presents time profiles for two representative medium-sized instances, Erding and Ring 5x5. The plots decompose total wall-clock time into three components: master problem solve time, subproblem solve time, and overhead. Overhead includes all computational effort not directly associated with solving an LP or MIP, such as data updates, cut generation, and solution aggregation on the root process.

As indicated by the leftmost bar in each plot, subproblem solve time dominates the serial implementation, accounting for 62.5% and 73.1% of total wall-clock time for Erding and Ring 5x5, respectively. This highlights substantial potential for computational gains through reducing subproblem solution time, which initially motivated the parallelization of subproblem solves.

Moreover, master problem solve time is effectively absent from the time profile plots, indicating that it is negligible. Even for the largest datasets, total master problem solution time is on the order of seconds, compared to minutes for subproblem solves and overhead. This disparity is driven by the imbalance in problem size between the upper- and lower-level decision spaces, as reflected in the dimensions reported in Table 2. For example, the Erding instance contains only 66 upper-level decision variables (one for each line $\ell \in \mathcal{L}$), whereas the lower-level problem involves nearly 3 million variables. Additionally, because the line-planning cost formulation generates feasibility cuts in most iterations, only a single constraint is added per outer Benders iteration, keeping the master problem relatively small.

The time profiles at higher processor counts demonstrate that the nested Benders framework effectively exploits the separability of the subproblems. Subproblem solve time – shown in green – decreases substantially as the number of processors increases, driving a corresponding reduction in total wall-clock time. For both Erding and Ring 5x5, subproblem time is reduced to the point where wall time is dominated by overhead at $p = 64$, accounting for 73.7% and 67.0% of total time, respectively.

Overhead also decreases with additional processors, largely due to the distribution of subproblem-specific tasks. However, this reduction eventually plateaus once $p \geq 64$, as a portion of the overhead – such as master problem updates and cut

				Serial ($p = 1$)		Parallel ($p = 64$)			
Dataset	$\lvert\mathcal{F}\rvert$	Iter.	Baseline (s)	Wall (s)	CPU (s)	Wall (s)	CPU (s)	Speedup	CPU Eff.
Ring 3×3	90	5	3.6	8.3	4.3	0.5	2.7	15.70	8.0%
Mandl	172	1	17.7	1.6	0.9	0.1	0.5	11.68	6.1%
Ring 4×4	272	71	547.6	1,236.7	681.3	56.1	532.6	22.04	14.8%
Sioux Falls	552	6	6,325.6	69.8	41.0	2.3	21.5	29.79	14.3%
Ring 5×5	650	28	1,773.9	5,350.3	3,735.5	158.2	2,539.2	33.82	25.1%
Erding	675	3	23,244.1	79.9	51.4	1.9	23.9	41.37	19.3%
Ring 6×6	1332	1	—	4,183.5	3,932.9	23.4	918.5	178.75	61.3%
Ring 7×6	1806	1	—	9,877.7	9,322.5	49.0	2,026.4	201.43	64.6%
Athens	2385	38	26,103.5	6,659.0	4,511.4	216.2	3,020.1	30.80	21.8%
Bahn	6106	1	—	3,448.9	2,813.7	57.3	1,654.9	60.16	45.1%

Table 3. Wall time $T(p)$ and CPU time $T_{\text{CPU}}(p)$ on $p = 64$ MPI processors compared to monolithic baseline on line-planning cost formulation (41). Speedup is computed by $S(p) = T(1)/T(p)$ and CPU efficiency by $E_{\text{CPU}}(p) = T_{\text{CPU}}(p)/(p \cdot T(p))$.

generation – remains inherently serial. This behavior reflects Amdahl's law: overall speedup is ultimately constrained by these serial components.

6.3.3. *Performance Against Baseline.* Table 3 reports wall-clock and CPU times for the nested Benders scheme applied to the line-planning cost formulation across all 10 instances, alongside a monolithic baseline. The baseline uses Gurobi 13.0.0 to solve the MINLP (7) directly, without decomposition. For the three largest datasets – Ring 6x6, Ring 7x6, and Bahn – the baseline fails to solve the MINLP within the 12-hour wall-clock time limit.

With the exception of the Ring 3x3, 4x4, and 5x5 instances, the serial nested Benders implementation achieves lower wall-clock times than the baseline. The parallel implementation with $p = 64$ processors outperforms the baseline on all datasets, often by orders of magnitude. These results demonstrate that the nested Benders decomposition significantly improves computational performance relative to the monolithic MINLP approach, on these large-scale, nonconvex instances. Recall that the monolithic MINLP approach solves (7) directly, with nonconvexity arising from the bilinear terms in the strong-duality constraints (7e).

Table 3 also reports speedup and CPU efficiency for $p = 64$ processors. The reported speedups $T(64)/T(1)$ correspond to those shown in Figure 6 (right). The largest speedups occur for the datasets Ring 6x6, Ring 7x6, and Bahn, driven by their convergence in a single Benders iteration.

For the remaining instances, speedups range from 11.68× (Mandl) to 41.37× (Erding), demonstrating substantial computational gains from parallelization. Relative to the baseline MINLP approach, these improvements are even more pronounced, with speedups ranging from 7.2× for the small Ring 3x3 instance up to 12,233.7× for the Erding dataset.

As defined in Section 6.2.2, CPU efficiency measures how effectively computational resources are used, with $E_{\text{CPU}}(p) = 100\%$ indicating perfect utilization. This efficiency typically mirrors the observed speedup, with the large datasets – Ring 6x6, Ring 7x6, and Bahn – achieving the highest efficiency due to convergence in a single Benders iteration. These instances are outliers, as they avoid many of the serial bottlenecks inherent to the algorithm.

The remaining instances exhibit CPU efficiencies ranging from 6.1% (Mandl) to 25.1% (Ring 5x5), with smaller datasets generally attaining lower efficiency. This behavior is expected, as communication overhead and limited per-process workload reduce effective resource utilization on small datasets. More broadly, CPU efficiency is constrained by inherently serial components of the algorithm, such as

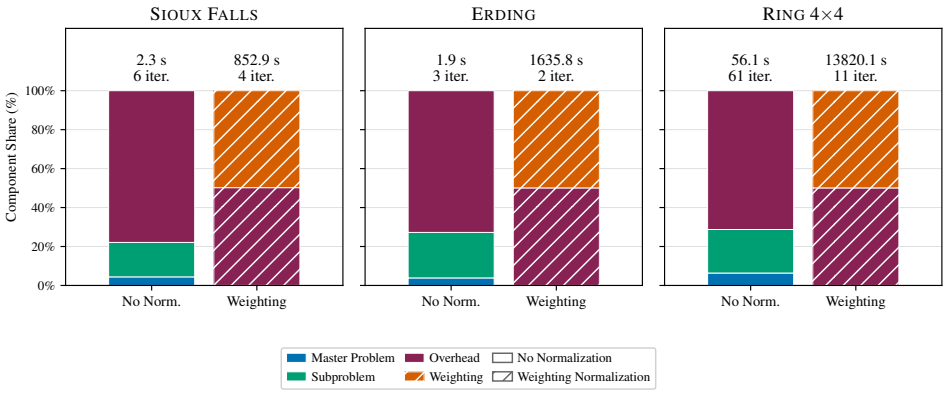


FIGURE 8. Component share (%) of wall time with and without unbounded dual ray weighting normalization on $p = 64$ processors.

cut generation and master problem updates at each iteration. Nonetheless, despite moderate efficiency, the parallel implementation delivers substantial reductions in wall-clock time, demonstrating that the available computational resources are effectively leveraged in practice.

6.3.4. *Comparison of Normalization Approaches.* Next, we consider the same LPMT instances, now incorporating the weighting normalization scheme introduced in Section 3.2. Recall that this scheme takes the nonzero extreme ray directions computed by the subproblem and solves the integrated LP (33) to reweight these multipliers so that their ℓ_1-norm equals 1. Following the feasibility cut generation framework of Seo et al. (2022), the objective of this normalization is to produce tighter (i.e., closer) feasibility cuts, thereby reducing the number of outer Benders iterations required for convergence while preserving the separability of the subproblems. Notably, as illustrated in Figure 5, solving the weighting problem introduces an additional serial task on the root processor.

Figure 8 presents time profiles for three representative datasets, comparing performance with and without the unbounded dual ray weighting normalization scheme on $p = 64$ processors. The total wall time and number of outer iterations required for convergence are reported above each bar. Applying the weighting scheme consistently reduces the number of outer Benders iterations, indicating that the weighted dual rays yield tighter feasibility cuts. However, this improvement comes at the cost of a substantial increase in overall wall time for the nested Benders procedure. In the most extreme case, the Ring 4x4 dataset sees a reduction in iterations from 61 to 11, but the corresponding wall time increases dramatically from 56.1 seconds without weighting to 13,820.1 seconds.

The increase in wall time is driven by the introduction of a serial bottleneck associated with the weighting problem. When the weighting normalization scheme is applied, solving the LP (33) accounts for roughly 50% of the total wall time on all three datasets shown in Figure 8. In addition, the pre- and post-processing required to construct and integrate the weighting problem introduces substantial overhead. As a result, the overall runtime becomes dominated by these components, effectively negating the benefits of parallelizing the subproblem solves.

By filtering for nonzero dual ray components, the intent was to keep the weighting problem relatively small compared to the full subproblem. In practice, however, the reduction was limited for these instances. Across the three datasets in Figure 8, the

				Serial ($n_p = 1$)		Parallel ($n_p = 16$)			
Dataset	Nr. Foll.	Iter.	Baseline (s)	Wall (s)	CPU (s)	Wall (s)	CPU (s)	Speedup	CPU Eff.
Ring 3×3	90	76	6.2	75.2	342.6	62.8	400.6	1.20	39.9%
Mandl	172	18	135.9	34.2	37.0	4.4	20.1	7.82	28.7%
Ring 4×4	272	137	256.9	2,455.1	34,049.9	3,073.6	22,449.1	0.80	45.6%
Sioux Falls	552	163	3,785.2	8,128.1	85,759.3	12,951.9	94,420.1	0.63	45.6%

TABLE 4. Wall time $T(p)$ and CPU time $T_{\text{CPU}}(p)$ on $p = 16$ MPI processors compared to monolithic baseline on total travel time formulation (43). Speedup is computed by $S(p) = T(1)/T(p)$ and CPU efficiency by $E_{\text{CPU}}(p) = T_{\text{CPU}}(p)/(p \cdot T(p))$.

average proportion of nonzero multipliers per outer iteration is approximately 20% (19.5% for Sioux Falls, 21.2% for Erding, and 20.6% for Ring 4x4).

When scaled against the number of lower-level constraints M_y, this still yields extremely large weighting problems, with between 250,000 and 650,000 decision variables that must be solved serially. Consequently, the anticipated benefit of reducing the number of outer Benders iterations is more than offset by the computational burden of solving these large-scale LPs at the root node. While this behavior may be problem-specific, in our setting the normalization scheme proves ineffective in practice.

6.4. **Total Travel Time Formulation.** Finally, we apply the nested Benders procedure to the total travel time formulation (43) described in Section 6.1.2. The motivation for considering this alternative formulation stems from two observations. First, the line-planning cost model (42) grants substantial control to the line planner (upper-level decision-maker) who does not directly accounting for the travel time ultimately experienced by passengers in their objective. This discrepancy can be seen in the line concepts computed by each formulation shown in Appendix B for the Ring 3x3 and 4x4 datasets. Second, because the upper-level objective in (42) does not contain lower-level variables (i.e., $c_f = 0$ for all $f \in \mathcal{F}$), convergence requires only feasibility cuts for the line cost formulation. Accordingly, the purpose of this analysis is twofold: to evaluate the generation of optimality cuts within the proposed framework and to compute solutions that are more passenger-centric. For these experiments, we set the line-planning budget Δ to 150% of the minimum cost obtained from the line-planning formulation for each respective dataset.

The total travel time formulation proved to be significantly more computationally challenging to solve within the nested Benders framework. This increased difficulty was driven primarily by substantial master problem solve times, which introduced a serial bottleneck into the overall procedure. As a result, we report results only for the four smallest datasets; the remaining instances did not converge within the prescribed 12-hour wall-clock time limit under the current implementation. In what follows, we analyze the factors contributing to the computational difficulty of solving this formulation using the proposed framework.

Table 4 reports the wall clock and CPU times of the nested Benders scheme on $p = 1$ and $p = 16$ processors for the four datasets, alongside the corresponding monolithic baseline results. The nested Benders scheme achieved a faster wall time than the baseline in both the serial and parallel implementations for only one dataset (Mandl). Moreover, in contrast to the results for the line-planning cost formulation, the parallel implementation produced only modest speedups and, in some cases, performed worse when additional processors were used. Mandl – the only dataset for which the nested Benders scheme outperformed the baseline – was also the sole instance to achieve substantial parallel acceleration, with a speedup of 7.82x on

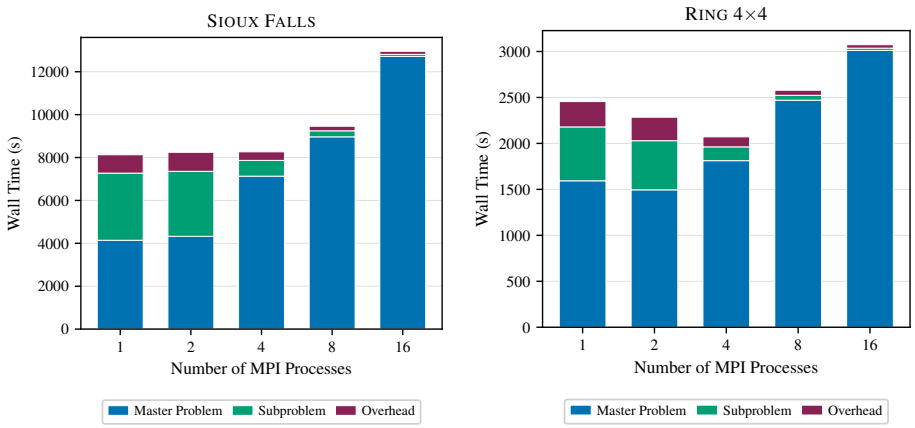


FIGURE 9. Walltime (s) breakdown for total travel time formulation on (Left) Sioux Falls and (right) Ring 4x4 datasets.

$p = 16$ processors. For the Sioux Falls and Ring 4x4 datasets, the reported wall times were actually higher on $p = 16$ processors than on a single processor.

Interestingly, the CPU efficiency, which how effectively computational resources are used, exceeded 28% for all four datasets on $p = 16$ processors, which is comparable to the efficiency observed for the line-planning cost formulation. This suggests that the implementation is effectively utilizing the available computational resources. Consequently, the high wall times point instead to the presence of a significant serial bottleneck within the overall procedure.

6.4.1. *Master Problem Wall Time.* This hypothesis that the higher wall times are caused by the presence of a significant serial bottleneck is confirmed by the wall time component shares shown in Figure 9 for two representative datasets. Similar to the line-planning cost formulation, the subproblem solve time and communication overhead decrease as additional processors are introduced. In contrast to the line-planning cost formulation, however, master problem solve time accounts for a substantial portion of the total wall time and grows in relative importance as the number of processors increases.

Specifically, master problem solves comprised 50.9% and 64.9% of the total wall time on $p = 1$ processor for Sioux Falls and Ring 4x4, respectively, compared to 98.2% and 98.0% on $p = 16$ processors. This shift was driven partly by the reduction in subproblem solve time, but also by an increase in overall wall-clock time of 59.2% and 25.2%, respectively, when moving from $p = 1$ to $p = 16$ processors.

The increase in wall time spent solving the master problem as additional processors are added is likely attributable to a "starved" master problem. As discussed in Section 6.2.1, each compute node used in the experiments contains 128 cores, and threads are assigned dynamically such that each MPI process receives $128/p$ threads. For example, when $p = 2$, each MPI process is allocated 64 threads, whereas when $p = 16$, each process receives only 8 threads. These threads enable the Gurobi solver to parallelize its branch-and-bound procedure for the master problem solve.

Because the master problem is always solved on the root process, the fastest master problem solves occur when the root process has access to the largest number of threads (i.e., the $p = 1$ or $p = 2$ case, where 128 or 64 threads are available). Consequently, increasing the number of processors improves the efficiency of the distributed subproblem solves while simultaneously reducing the computational

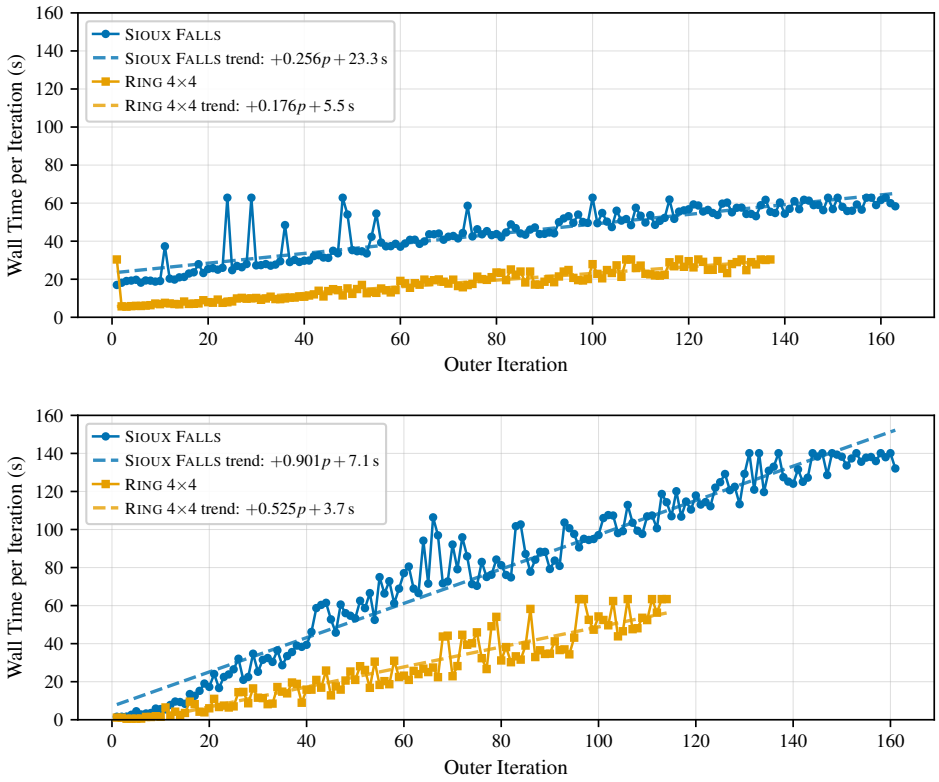


FIGURE 10. Wall time (s) per outer Benders iteration for total travel time formulation of Sioux Falls and Ring 4x4 datasets on (top) 1 processor, 128 threads and (bottom) 16 processors, 8 threads.

resources available for solving the master problem at the root node. In the line-planning cost formulation, this effect was negligible because the master problem was relatively inexpensive to solve compared to the subproblems.

6.4.2. *Growing Master Problem Size.* As described in the overviews of feasibility and optimality cut generation in Sections 3.1.1 and 3.1.2, a key distinction between the two approaches is that only a single feasibility cut is generated per iteration, whereas up to $|\mathcal{F}|$ optimality cuts may be added in each iteration. The optimality cuts employ the disaggregated multi-cut generation strategy described in Section 5.3. The motivation for this approach was to reduce the number of iterations required for convergence relative to a scheme that adds only a single aggregated optimality cut per iteration. The tradeoff, however, is that the number of constraints in the master problem grows substantially as the algorithm progresses.

Figure 10, which plots the wall time per outer Benders iteration for two representative datasets that exhibited longer wall times on $p = 16$ processors than on $p = 1$ processor, illustrates this phenomenon clearly. For both datasets, the time per iteration increases steadily as the iteration count progresses. During the early iterations, the iteration times remain relatively manageable; however, they subsequently increase at rates of 0.256s and 0.176s per iteration for the Sioux Falls and Ring 4x4 datasets, respectively, on $p = 1$ processor.

This per-iteration growth is further exacerbated in the parallel implementation on $p = 16$ processors, where the master problem has access to only 8 threads. In this

setting, the master problem solve time increases at rates of 0.901s and 0.525s per iteration for Sioux Falls and Ring 4x4, respectively. Across the full solve, 30,625 and 17,352 optimality cuts are added over 163 and 137 iterations for the Sioux Falls and Ring 4x4 datasets, respectively, resulting in increasingly expensive master problem solves as the algorithm progresses.

Overall, the current nested Benders framework appears presents challenges for solving the total time formulation. While the disaggregated multi-cuts successfully reduce the number of outer Benders iterations required for convergence, they also lead to a growing master problem that can introduce a serial bottleneck, offsetting the benefits of parallel subproblem solves. There exist several advanced techniques for managing master problem size in Benders decomposition, including age-based cut pruning and slack-based cut analysis. Incorporating such strategies to improve master problem tractability within this framework represents a promising direction for future research.

7. Conclusions

We propose a scalable nested Benders decomposition (BD) algorithm for solving single-leader, multi-follower games. The framework applies to problems in which each follower solves a linear program and is particularly effective for instances with many followers. The method exploits the upper-level decision variables as complicating variables and leverages the separable structure of the lower-level problems, which decompose by follower once the upper-level decisions are fixed. The resulting algorithm combines an outer and inner BD scheme, incorporates closest Benders cuts following Seo et al. (2022) to reduce the number of outer iterations, and employs parallel computation to accelerate the solution of the inner subproblems.

The nested BD framework is implemented on a distributed-memory architecture to enable scalable parallel execution. The algorithm uses a hybrid parallelization strategy: MPI is employed for inter-processor communication across distributed memory, while OpenMP is used for shared-memory parallelization within each processor, particularly for computationally intensive linear algebra operations such as matrix multiplication in model formulations and subproblem updates. The result is an implementation and parallel framework optimized for HPC environments, enabling the framework to be solve large scale models.

We apply the approach to two formulations of a line planning problem with integrated passenger routing across 10 public transportation networks containing up to 51 million variables and 128 million constraints. The method proves particularly effective for the line planning cost formulation, in which feasibility cuts are generated at most outer Benders iterations and the dominant computational effort lies in solving the subproblems rather than the master problem. The parallel implementation demonstrates strong scaling behavior on these large-scale instances, substantially accelerating the subproblem solves, achieving significant speedups as the number of processors increases, and significantly outperforming the monolithic MINLP baseline.

Another advantage of the proposed framework is its compatibility with advanced cut generation and master problem size-management strategies. For example, the disaggregated multi-cut approach of Laporte and Louveaux (1993) is adapted for optimality cut generation and proves effective at reducing the number of iterations required for convergence. Nevertheless, the algorithm is less effective for the total travel time formulation of the bilevel LPMT, where the dominant computational burden shifted to the master problem and convergence depends heavily on the accumulation of optimality cuts. In these instances, incorporating more advanced

size-management techniques, such as age-based cut pruning and slack analysis, could significantly improve performance.

Furthermore, sophisticated strategies, including branch-and-Benders-cut implementations using lazy constraints (Papadakos 2008), are naturally compatible with the proposed framework and may further enhance computational efficiency on challenging instances. Overall, while the proposed algorithm is highly effective at reducing subproblem solution times through parallelization and the use of available computational resources, improving master problem solution times and alleviating the resulting serial bottlenecks remain important directions for future research.

Acknowledgments

The authors acknowledge the University of Maryland supercomputing resources (https://hpcc.umd.edu) made available for conducting the research reported in this paper. S. Gabriel and D. Flocco were supported by a grant from Petrobras #4324713. D. Flocco acknowledges support from the Aalto Science Institute Visiting Doctoral Researcher Program. P. Schiewe was supported by the Research Council of Finland (Flagship of Advanced Mathematics for Sensing Imaging and Modelling 359181).

References

Adulyasak, Y., J.-F. Cordeau, and R. Jans (2015). "Benders Decomposition for Production Routing Under Demand Uncertainty." In: *Operations Research* 63.4, pp. 851–867. DOI: 10.1287/opre.2015.1401.

Alipour, M., K. Zare, and H. Seyedi (2018). "A multi-follower bilevel stochastic programming approach for energy management of combined heat and power microgrids." In: *Energy* 149, pp. 135–146. DOI: 10.1016/j.energy.2018.02.013.

Amdahl, G. M. (1967). "Validity of the single processor approach to achieving large scale computing capabilities." In: *Proceedings of the AFIPS Spring Joint Computer Conference*. ACM, pp. 483–485. DOI: 10.1145/1465482.1465560.

Askeland, M., T. Burandt, and S. A. Gabriel (2023). "A stochastic MPEC approach for grid tariff design with demand-side flexibility." In: *Energy Systems* 14.3, pp. 707–729. DOI: 10.1007/s12667-020-00407-7.

Bailly, G., M. Cornet, M. Glavic, and B. Cornélusse (2023). "A one-leader multi-follower approach to distribution network development planning." In: *2023 IEEE PES Innovative Smart Grid Technologies Europe (ISGT EUROPE)*, pp. 1–5. DOI: 10.1109/ISGTEUROPE56780.2023.10408702.

Bard, J. F. (1991). "Some properties of the bilevel programming problem." In: *Journal of Optimization Theory and Applications* 68.2, pp. 371–378. DOI: 10.1007/BF00941574.

– (1998). *Practical Bilevel Optimization: Algorithms and Applications*. Vol. 30. Springer Science & Business Media.

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance (1998). "Branch-and-Price: Column Generation for Solving Huge Integer Programs." In: *Operations Research* 46.3, pp. 316–329. DOI: 10.1287/opre.46.3.316.

Basilico, N., S. Coniglio, N. Gatti, and A. Marchesi (2020). "Bilevel programming methods for computing single-leader-multi-follower equilibria in normal-form and polymatrix games." In: *EURO Journal on Computational Optimization* 8.1, pp. 3–31. DOI: 10.1007/s13675-019-00114-8.

Beheshti, B., O. A. Prokopyev, and E. L. Pasiliao (2016). "Exact solution approaches for bilevel assignment problems." In: *Computational Optimization and Applications* 64.1, pp. 215–242. DOI: 10.1007/s10589-015-9799-4.

Benders, J. F. (Dec. 1962). "Partitioning procedures for solving mixed-variables programming problems." In: *Numerische Mathematik* 4.1, pp. 238–252. DOI: 10.1007/BF01386316.

Bertsimas, D. and J. N. Tsitsiklis (1997). *Introduction to Linear Optimization*. Vol. 6. Belmont, MA: Athena Scientific.

Birge, J. R. and F. V. Louveaux (1988). "A multicut algorithm for two-stage stochastic linear programs." In: *European Journal of Operational Research* 34.3, pp. 384–392. DOI: 10.1016/0377-2217(88)90159-2.

Borges, P., C. Sagastizábal, and M. Solodov (2021). "Decomposition Algorithms for Some Deterministic and Two-Stage Stochastic Single-Leader Multi-Follower Games." In: *Computational Optimization and Applications* 78.3, pp. 675–704. DOI: 10.1007/s10589-020-00257-0.

Cerulli, M., C. Archetti, E. Fernández, and I. Ljubić (2024). "A bilevel approach for compensation and routing decisions in last-mile delivery." In: *Transportation Science* 58.5, pp. 1076–1100. DOI: 10.1287/trsc.2023.0129.

Chermakani, D. P. (2015). "Optimal Aggregation of Blocks into Subproblems in Linear-Programs with Block-Diagonal-Structure." In: *arXiv preprint arXiv:1507.05753*.

Conejo, A. J., E. Castillo, R. Minguez, and R. Garcia-Bertrand (2006). *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*. Springer.

Cordeau, J.-F., G. Stojković, F. Soumis, and J. Desrosiers (2001). "Benders decomposition for simultaneous aircraft routing and crew scheduling." In: *Transportation Science* 35.4, pp. 375–388.

Dantzig, G. B. and P. Wolfe (1960). "Decomposition Principle for Linear Programs." In: *Operations Research* 8.1, pp. 101–111. DOI: 10.1287/opre.8.1.101.

Dempe, S. (2002). *Foundations of Bilevel Programming*. Vol. 61. Nonconvex Optimization and Its Applications. Boston, MA: Springer US. DOI: 10.1007/b101970.

Dempe, S., V. Kalashnikov, J. Perez-Valdes, and N. Kalashnykova (2015). *Bilevel Programming Problems*. Springer.

Dimanchev, E., S. A. Gabriel, S.-E. Fleten, F. Pecci, and M. Korpås (2024). "Choosing climate policies in a second-best world with incomplete markets: Insights from a bilevel power system model." In: *Energy Economics* 138, p. 107865. DOI: 10.1016/j.eneco.2024.107865.

Fischetti, M., D. Salvagnin, and A. Zanette (July 2010). "A note on the selection of Benders' cuts." In: *Mathematical Programming* 124.1, pp. 175–182. DOI: 10.1007/s10107-010-0365-7.

Fontaine, P. and S. Minner (2014). "Benders Decomposition for Discrete–Continuous Linear Bilevel Problems with application to traffic network design." In: *Transportation Research Part B: Methodological* 70, pp. 163–172. DOI: 10.1016/j.trb.2014.09.007.

Fortuny-Amat, J. and B. McCarl (1981). "A Representation and Economic Interpretation of a Two-Level Programming Problem." In: *The Journal of the Operational Research Society* 32.9, pp. 783–792. DOI: 10.2307/2581394.

Fortz, B. and M. Poss (2009). "An improved Benders decomposition applied to a multi-layer network design problem." In: *Operations Research Letters* 37.5, pp. 359–364. DOI: 10.1016/j.orl.2009.05.007.

Fukuda, K. and A. Prodon (1996). "Double description method revisited." In: *Combinatorics and Computer Science*. Springer Berlin Heidelberg, pp. 91–111. DOI: 10.1007/3-540-61576-8_77.

Fuller, J. D. and W. Chung (June 2005). "Dantzig–Wolfe Decomposition of Variational Inequalities." In: *Computational Economics* 25.4, pp. 303–326. DOI: 10.1007/s10614-005-2519-x.

– (2008). "Benders decomposition for a class of variational inequalities." In: *European Journal of Operational Research* 185.1, pp. 76–91.

Gabriel, S. A. and J. D. Fuller (Apr. 2010). "A Benders Decomposition Method for Solving Stochastic Complementarity Problems with an Application in Energy." In: *Computational Economics* 35.4, pp. 301–329. DOI: 10.1007/s10614-010-9200-8.

Gabriel, S. A., A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz (2012). *Complementarity Modeling in Energy Markets*. Vol. 180. Springer Science & Business Media.

Gabriel, S. A., M. Leal, and M. Schmidt (2022). "On linear bilevel optimization problems with complementarity-constrained lower levels." In: *Journal of the Operational Research Society* 73.12, pp. 2706–2716. DOI: 10.1080/01605682.2021.2015254.

Gelareh, S., R. Neamatian Monemi, and S. Nickel (2015). "Multi-period hub location problems in transportation." In: *Transportation Research Part E: Logistics and Transportation Review* 75, pp. 67–94. DOI: 10.1016/j.tre.2014.12.016.

Goerigk, M. and M. Schmidt (2017). "Line planning with user-optimal route choice." In: *European Journal of Operational Research* 259.2, pp. 424–436.

Hager, G. and G. Wellein (2010). *Introduction to High Performance Computing for Scientists and Engineers*. Boca Raton, FL: CRC Press.

Han, J., J. Lu, Y. Hu, and G. Zhang (2015). "Tri-level decision-making with multiple followers: Model, algorithm and case study." In: *Information Sciences* 311, pp. 182–204. DOI: 10.1016/j.ins.2015.03.043.

Herrala, O., S. A. Gabriel, F. Oliveira, and T. Ekholm (2025). "A novel strong duality-based reformulation for trilevel infrastructure models in energy systems development." In: *Journal of the Operational Research Society* 76.3, pp. 438–453. DOI: 10.1080/01605682.2024.2365807.

Holmberg, K. (1990). "On the convergence of cross decomposition." In: *Mathematical Programming* 47.1, pp. 269–296.

Hosseini, M. and J. Turner (2024). *Deepest Cuts for Benders Decomposition*. arXiv: 2110.08448 [math.OC].

Hu, J. and D. Ralph (2007). "EPECs and multi-leader-follower games." In: *SIAM Journal on Optimization* 18.3, pp. 977–1002.

Huppmann, D. and J. Egerer (2015). "National-strategic investment in European power transmission capacity." In: *European Journal of Operational Research* 247.1, pp. 191–203. DOI: 10.1016/j.ejor.2015.05.056.

Kaltis, T. and G. K. D. Saharidis (2026). "Literature review on Benders cut selection and a multiple cut generation scheme." In: *INFOR: Information Systems and Operational Research* 64.1, pp. 244–270. DOI: 10.1080/03155986.2025.2540205.

Kleinert, T., M. Labbé, F. Plein, and M. Schmidt (2020). "Technical Note—There's No Free Lunch: On the Hardness of Choosing a Correct Big-M in Bilevel Optimization." In: *Operations Research* 68.4, pp. 1135–1142. DOI: 10.1287/opre.2019.1944.

Laporte, G. and F. V. Louveaux (1993). "The integer L-shaped method for stochastic integer programs with complete recourse." In: *Operations Research Letters* 13.3, pp. 133–142. DOI: 10.1016/0167-6377(93)90002-X.

Lei, C., Z. Jiang, and Y. Ouyang (2020). "Path-based dynamic pricing for vehicle allocation in ridesharing systems with fully compliant drivers." In: *Transportation*

Research Part B: Methodological 132. 23rd International Symposium on Transportation and Traffic Theory (ISTTT 23), pp. 60–75. DOI: 10.1016/j.trb.2019.01.017.

Lin, H. and H. Üster (2014). "Exact and Heuristic Algorithms for Data-Gathering Cluster-Based Wireless Sensor Network Design Problem." In: *IEEE/ACM Transactions on Networking* 22.3, pp. 903–916. DOI: 10.1109/TNET.2013.2262153.

Linderoth, J. and S. Wright (Feb. 2003). "Decomposition Algorithms for Stochastic Programming on a Computational Grid." In: *Computational Optimization and Applications* 24.2, pp. 207–250. DOI: 10.1023/A:1021858008222.

Luna, J. P., C. Sagastizábal, and M. Solodov (2020). "A class of Benders decomposition methods for variational inequalities." In: *Computational Optimization and Applications* 76.3, pp. 935–959. DOI: 10.1007/s10589-019-00157-y.

Luo, Z. Q., J. S. Pang, and D. Ralph (1996). *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press.

Mercier, A. and F. Soumis (2007). "An integrated aircraft routing, crew scheduling and flight retiming model." In: *Computers & Operations Research* 34.8, pp. 2251–2265. DOI: 10.1016/j.cor.2005.09.001.

Mitsos, A., P. Lemonidis, and P. I. Barton (2008). "Global optimization of bilevel programs with a nonconvex inner program." In: *Journal of Global Optimization* 42, pp. 475–513. DOI: 10.1007/s10898-007-9260-z.

Pacqueau, R., F. Soumis, and L. N. Hoang (2012). *A Fast and Accurate Algorithm for Stochastic Integer Programming, Applied to Stochastic Shift Scheduling*. Technical Report. GERAD, Montréal QC H3T 2A7, Canada: Groupe d'études et de recherche en analyse des décisions, pp. 1–19.

Pang, J.-S. and M. Fukushima (2005). "Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games." In: *Computational Management Science* 2.1, pp. 21–56.

Papadakos, N. (2008). "Practical enhancements to the Magnanti–Wong method." In: *Operations Research Letters* 36.4, pp. 444–449. DOI: 10.1016/j.orl.2008.01.005.

– (2009). "Integrated airline scheduling." In: *Computers & Operations Research* 36.1. Part Special Issue: Operations Research Approaches for Disaster Recovery Planning, pp. 176–195. DOI: 10.1016/j.cor.2007.08.002.

Rahmaniani, R., T. G. Crainic, M. Gendreau, and W. Rei (2017). "The Benders decomposition algorithm: A literature review." In: *European Journal of Operational Research* 259.3, pp. 801–817. DOI: 10.1016/j.ejor.2016.12.005.

Saharidis, G. K. D., M. Minoux, and M. G. Ierapetritou (2010). "Accelerating Benders method using covering cut bundle generation." In: *International Transactions in Operational Research* 17.2, pp. 221–237. DOI: 10.1111/j.1475-3995.2009.00706.x.

Sahinidis, N. V. and I. E. Grossmann (1991). "Convergence properties of generalized Benders decomposition." In: *Computers & Chemical Engineering* 15.7, pp. 481–491. DOI: 10.1016/0098-1354(91)85027-R.

Schiewe, P., A. Schöbel, O. Herrala, M. Rihlmann, S. Roth, S. Albert, C. Biedinger, T. Dahlheimer, L. Dittrich, K. Hoffmann, S. Jäger, P. Pattanaik, A. Schiewe, M. Stinzendörfer, and R. Urban (2025). *Documentation for LinTim 2025.11*. Tech. rep. Kaiserslautern – Fachbereich Mathematik, p. 225.

– (2026). *LinTim – Integrated Optimization in Public Transportation*. Open-source software. Accessed May 2026. URL: https://www.lintim.net/.

Schmidt, M. and A. Schöbel (2024). *Planning and Optimizing Transit Lines*. arXiv: 2405.10074 [math.OC].

Schöbel, A. (2012). "Line planning in public transportation: models and methods." In: *OR Spectrum* 34.3, pp. 491–510.

Schöbel, A. and S. Scholl (2006). "Line Planning with Minimal Traveling Time." In: *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*. Ed. by L. G. Kroon and R. H. Möhring. Vol. 2. Open Access Series in Informatics (OASIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 1–16. DOI: 10.4230/OASIcs.ATMOS.2005.660.

Seo, K., S. Joung, C. Lee, and S. Park (2022). "A closest Benders cut selection scheme for accelerating the Benders decomposition algorithm." In: *INFORMS Journal on Computing* 34.5, pp. 2804–2827. DOI: 10.1287/ijoc.2022.1207.

Siddiqui, S. and S. A. Gabriel (2013). "An SOS1-based approach for solving MPECs with a natural gas market application." In: *Networks and Spatial Economics* 13.2, pp. 205–227.

U-tapao, C., S. Moryadee, S. A. Gabriel, C. Peot, and M. Ramirez (2016). "A stochastic, two-level optimization model for compressed natural gas infrastructure investments in wastewater management." In: *Journal of Natural Gas Science and Engineering* 28, pp. 226–240. DOI: 10.1016/j.jngse.2015.11.039.

Van Slyke, R. M. and R. J.-B. Wets (1969). "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming." In: *SIAM Journal on Applied Mathematics* 17.4, pp. 638–663. DOI: 10.1137/0117061.

Vladimirou, H. (1998). "Computational assessment of distributed decomposition methods for stochastic linear programs." In: *European Journal of Operational Research* 108.3, pp. 653–670. DOI: 10.1016/S0377-2217(97)00222-1.

Wolf, C. (2014). "Advanced acceleration techniques for nested Benders decomposition in stochastic programming." PhD thesis. Universität Paderborn.

Wolf, C. and A. Koberstein (2013). "Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method." In: *European Journal of Operational Research* 230.1, pp. 143–156. DOI: 10.1016/j.ejor.2013.04.017.

Woo, Y.-B. and I. Moon (2025). "Bilevel optimization for multi-user systems with mixed demand response programs for enhanced operational efficiency in electric power grids." In: *Applied Energy* 399, p. 126507. DOI: 10.1016/j.apenergy.2025.126507.

Xi, H., D. Aussel, W. Liu, S. T. Waller, and D. Rey (2024). "Single-leader multi-follower games for the regulation of two-sided mobility-as-a-service markets." In: *European Journal of Operational Research* 317.3, pp. 718–736. DOI: 10.1016/j.ejor.2022.06.041.

Yang, Y. and J. M. Lee (2012). "A tighter cut generation strategy for acceleration of Benders decomposition." In: *Computers & Chemical Engineering* 44, pp. 84–93. DOI: 10.1016/j.compchemeng.2012.04.015.

Zakeri, G., A. B. Philpott, and D. M. Ryan (Jan. 2000). "Inexact Cuts in Benders Decomposition." In: *SIAM Journal on Optimization* 10.3, pp. 643–657. DOI: 10.1137/S1052623497318700.

Zare, M. H., J. S. Borrero, B. Zeng, and O. A. Prokopyev (2019). "A note on linearized reformulations for a class of bilevel linear integer problems." In: *Annals of Operations Research* 272.1, pp. 99–117. DOI: 10.1007/s10479-017-2694-x.

Zha, L., Y. Yin, and Y. Du (2017). "Surge Pricing and Labor Supply in the Ride-Sourcing Market." In: *Transportation Research Procedia* 23. Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory, Chicago, Illinois, USA, 24–26 July, 2017, pp. 2–21. DOI: 10.1016/j.trpro.2017.05.002.

Appendix A. Decomposable Structure of LPMT-BL

In this section, we make the application of the nested BD framework to the LPMT-BL more concrete. Specifically, we outline the decomposable structure of the strong-duality formulation of the bilevel problem (42).

A.1. **Routing Constraints.** The linear system in the lower-level represents the passenger routing constraints. Specifically, the matrix $\Theta \in \mathbb{Z}^{|\mathcal{V}|\times|\mathcal{E}|}$ is the node-arc-incidence matrix of the change & go network $\mathcal{G}$.

For a given OD pair $r = (s,t) \in \mathcal{R}$ in the LPMT model, these constraints are of the form

$$\sum_{e\in\delta^+(s,0)} y_r^e = 1 \tag{46a}$$

$$\sum_{e\in\delta^+(v)} y_r^e - \sum_{e\in\delta^-(v)} y_r^e = 0 \quad \forall v \in \mathcal{V}\backslash\{s,t\} \tag{46b}$$

$$-\sum_{e\in\delta^-(t,0)} y_r^e = -1, \tag{46c}$$

where $\delta^-(v)$ and $\delta^+(v)$ are the set of incoming and outgoing edges of node $v \in \mathcal{V}$, respectively. Therefore, the entry in row (node) i and column (edge) j of matrix Θ is

$$\Theta[i,j] = \begin{cases} 1 & \text{if } j \in \delta^+(i) \\ -1 & \text{if } j \in \delta^-(i) \\ 0 & \text{otherwise.} \end{cases} \tag{47}$$

And the entry i in the vector b_r is

$$b_r[i] = \begin{cases} 1 & \text{if } i = (s,0) \\ -1 & \text{if } i = (t,0) \\ 0 & \text{otherwise.} \end{cases} \tag{48}$$

A.2. **Strong Duality Formulation.** The relaxed lower-level problem of (LPMT-BL) for OD pair $r = (s,t) \in \mathcal{R}$ is as follows:

$$\underset{y_r}{\text{minimize}} \quad \sum_{e\in\mathcal{E}} d_e y_r^e \tag{49a}$$

$$\text{s.t.} \quad \Theta y_r = b_r : \quad \lambda_r \tag{49b}$$

$$y_r^e \le x_\ell \quad : \quad \nu_{r,\ell}^e \qquad \forall \ell \in \mathcal{L}, e \in \mathcal{E}^\ell \tag{49c}$$

$$0 \le y_r^e \qquad \forall e \in \mathcal{E}. \tag{49d}$$

Note that (49c) implicitly enforces $y_r^e \le 1$ for the relaxation since $x_\ell \in \{0,1\}$. Here (49c) is the linking constraint that is also a function of the upper-level decision x_ℓ. The dual to (49) is

$$\underset{\nu,\lambda}{\text{maximize}} \quad b_r^\top \lambda_r - \sum_{\ell\in\mathcal{L}} x_\ell \sum_{e\in\mathcal{E}^\ell} \nu_{r,\ell}^e \tag{50a}$$

$$\text{s.t.} \quad [\Theta^\top \lambda_r]_e - \sum_{\ell\in\mathcal{L}: e\in\mathcal{E}^\ell} \nu_{r,\ell}^e \le d_e \qquad \forall e \in \mathcal{E} \tag{50b}$$

$$\nu_{r,\ell}^e \ge 0 \qquad \forall \ell \in \mathcal{L}, e \in \mathcal{E}^\ell. \tag{50c}$$

Based on the definition of the matrix Θ in (47), the matrix-vector product in dual constraint (50b) has the following form:

$$\left[\Theta^\top \lambda_r\right]_e = \lambda_r^u - \lambda_r^v \qquad \text{for } e = (u,v). \tag{51}$$

We can express the optimality conditions of (49) through the three conditions: strong duality, primal feasibility and dual feasibility. Weak duality with respect to (49) and (50) states:

$$b_r^\top \lambda_r - \sum_{\ell \in \mathcal{L}} x_\ell \sum_{e \in \mathcal{E}^\ell} \nu_{r,\ell}^e \leq \sum_{e \in \mathcal{E}} d_e y_r^e, \tag{52}$$

for all dual feasible λ_r, ν_r and η_r and primal feasible y_r. Hence, strong duality implies that (52) holds at equality. We can equivalently express strong duality by reversing the inequality in the weak duality condition (52), since the converse always holds, and refer to this a "reverse" weak duality. Then, y_r^* and $(\lambda_r^*, \nu_r^*, \eta_r^*)$ are optimal to (49) and (50), respectively, if and only if the following conditions hold:

$$b_r^\top \lambda_r - \sum_{\ell \in \mathcal{L}} x_\ell \sum_{e \in \mathcal{E}^\ell} \nu_{r,\ell}^e \geq \sum_{e \in \mathcal{E}} d_e y_r^e \tag{53a}$$

$$(49b), (49c), (49d) \tag{53b}$$

$$(50b), (50c). \tag{53c}$$

Note that all constraints in (53) are linear, except for (53a) which has a bilinear term. This arises from the product of upper-level decision x_ℓ and lower-level dual ν. If we define auxiliary variable $\nu_{r,\ell} = \sum_{e \in \mathcal{E}} \nu_{r,\ell}^e$, then the bilinear term is $x_\ell \cdot \nu_{e,\ell}$.

A.3. **Decomposition of LPMT-BL.** Consider the bilevel line-planning problem LPMT-BL (42). The above sections discuss single-level reformulations through either the concatenation of optimality conditions of the lower-level problems, of which there are $|\mathcal{R}|$, one for each OD-pair. This can be done through the strong duality formulation. Primal feasibility for the lower-level problems (42c) state

$$\Theta y_r = b_r, \;\; Px + Qy_r \leq 0, \qquad \forall r \in \mathcal{R} \tag{54}$$

where Θ is the $|\mathcal{V}| \times |\mathcal{E}|$ node-arc-incidence matrix for the change & go network $\mathcal{G}$ and P, Q are suitably defined matrices to compute the capacity constraints $y_r^e \leq x_\ell$ for all $\ell \in \mathcal{L}, e \in \mathcal{E}^\ell$. Also, $x \in \{x_\ell\}_{\ell \in \mathcal{L}}$ is the vector of the upper-level decisions, which are exogenous to the lower-level players. When we consider the primal feasibility constraints (54), we note that the resulting system of linear constraints is separable for each r. Denote by $n_r = |\mathcal{R}|$ the number of OD-pairs in the PTN. We can capture the set of all primal feasibility constraints with the following linear systems:

$$\begin{bmatrix} \Theta & 0 & \cdots & 0 \\ 0 & \Theta & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & & \Theta \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_r} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_r} \end{bmatrix}, \qquad \begin{bmatrix} P & Q & 0 & \cdots & 0 \\ P & 0 & Q & \cdots & 0 \\ \vdots & & \ddots & \vdots & \\ P & 0 & \dots & & Q \end{bmatrix} \begin{bmatrix} x \\ y_1 \\ y_2 \\ \vdots \\ y_{n_r} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{55}$$

Written this way, the set of lower-level problems are completely decomposable to each OD-pair. The decisions are linked in the upper-level by the capacity constraint (42b). In matrix form, we can write these linking constraints as

$$Tx + \sum_{r \in \mathcal{R}} W_r y \leq 0, \tag{56}$$

where $T = |\mathcal{R}| \text{diag}(|\mathcal{E}^1|, \dots, |\mathcal{E}^{|\mathcal{L}|}|)$ and W_r is a $|\mathcal{L}| \times |\mathcal{E}|$ matrix. Then the combined upper-level and lower-level primal feasibility constraints can be expressed by the

following linear systems:

$$\begin{bmatrix} T & W_1 & W_2 & \cdots & W_{n_r} \\ 0 & \Theta & 0 & \cdots & 0 \\ \vdots & 0 & \Theta & & 0 \\ \vdots & 0 & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \Theta \end{bmatrix} \begin{bmatrix} x \\ y_1 \\ y_2 \\ \vdots \\ y_{n_r} \end{bmatrix} = \begin{bmatrix} 0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n_r} \end{bmatrix} \tag{57}$$

and the inequality constraints:

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ P & Q & 0 & \cdots & 0 \\ \vdots & 0 & Q & & 0 \\ \vdots & 0 & & \ddots & \vdots \\ P & 0 & 0 & 0 & Q \end{bmatrix} \begin{bmatrix} x \\ y_1 \\ y_2 \\ \vdots \\ y_{n_r} \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{58}$$

In this setting, we have both complicating constraints (i.e., the linking constraints (42b)) and complicating variables (upper-level variables x in the lower-level).

Appendix B. Line Concept Solutions

In this section, we provide a set of illustrative graphics and solution metrics on the bilevel line-planning problem formulations. Specifically, we report the line pool $\mathcal{L}$ and solution for the Ring 3x3 and Ring 4x4 datasets for the line-planning cost formulation (42) described in Section 6.1.1 and the total travel time formulation (43) described in Section 6.1.2.

Table 5 provides a summary of the two objectives, line planning cost computed by (42a) and total passenger travel time computed by (43a), at optimality for the two formulations across four datasets. Figures 11 and 13 show an illustration of the line pools $\mathcal{L}$ for the Ring 3x3 and Ring 4x4 datasets. Finally, Figures 12 and 14 show the optimal line concepts (i.e., the x_ℓ decisions) that correspond to the objective function values reported in Table 5. Figures 11, 12, 13 and 14 were produced using the LinTim software (Schiewe et al. 2025, 2026).

Dataset	**Formulation**	$\lvert\mathcal{L}\rvert$	**Open**	**Cost**	**Time**
Mandl	Line Planning Cost	19	4	203.4	359,264.7
	Total Travel Time	19	5	255.5	257,409.7
Sioux Falls	Line Planning Cost	50	4	203.0	80,782.5
	Total Travel Time	50	6	304.4	68,312.7
Ring 3×3	Line Planning Cost	21	4	207.1	74,593.4
	Total Travel Time	21	5	263.9	43,436.4
Ring 4×4	Line Planning Cost	48	4	213.9	358,489.4
	Total Travel Time	48	6	320.4	162,349.5

Table 5. Summary of solutions.

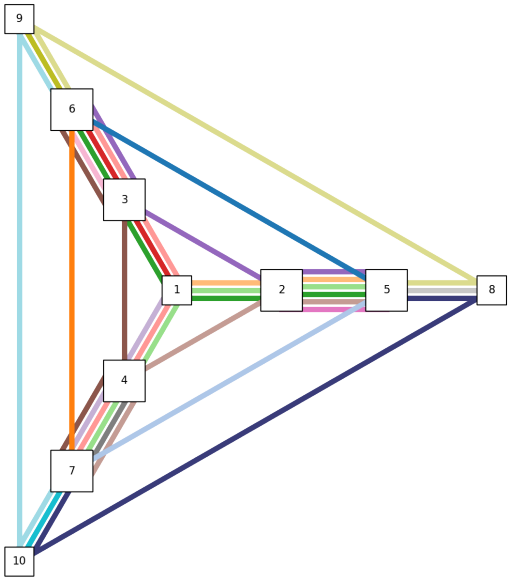


FIGURE 11. Line pool for Ring 3x3 dataset.

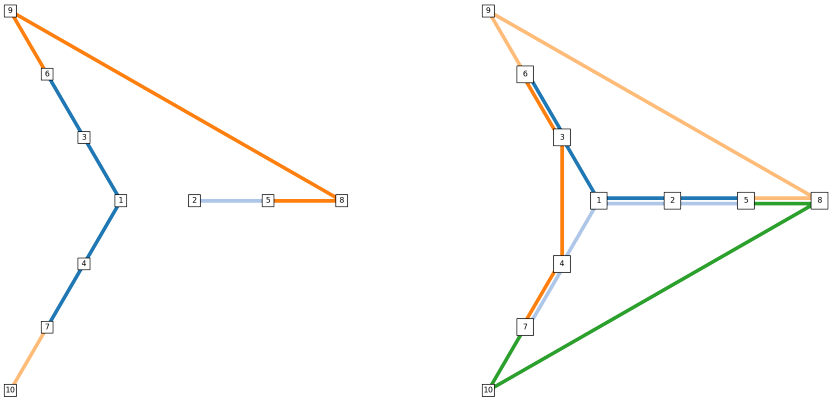


FIGURE 12. Optimal line concepts on Ring 3x3 dataset for (left) line planning cost and (right) total travel time formulations.

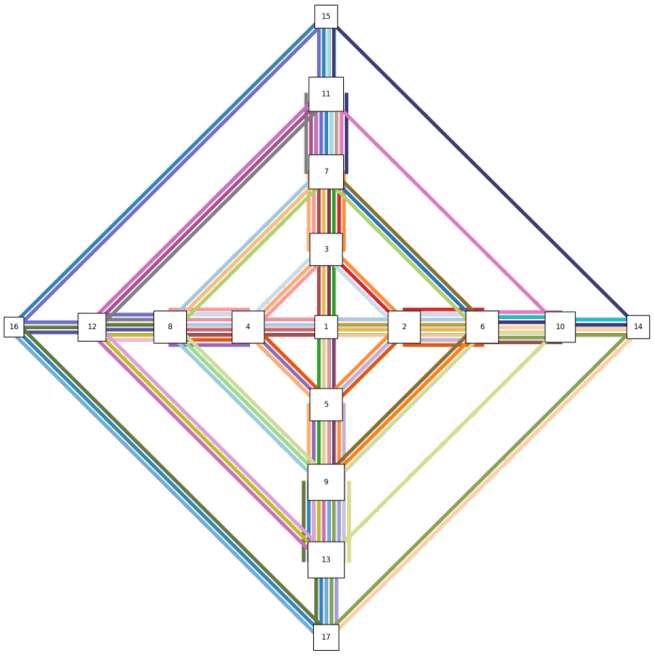


FIGURE 13. Line pool for Ring 4x4 dataset.

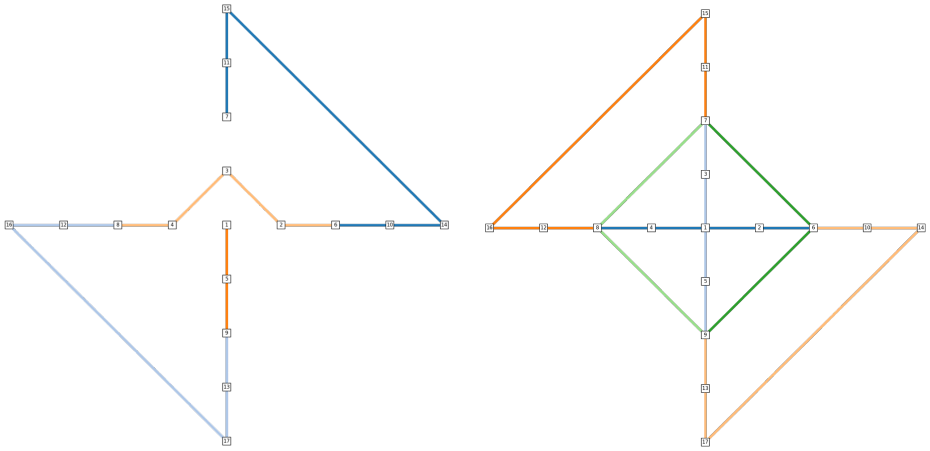


FIGURE 14. Optimal line concepts on Ring 4x4 dataset for (left) line planning cost and (right) total travel time formulations.

(D. Flocco) University of Maryland, Department of Mathematics, College Park, Maryland 20742, USA
Email address: dflocco@umd.edu

(P. Schiewe) Aalto University, Department of Mathematics and Systems Analysis, Espoo, Finland
Email address: philine.schiewe@aalto.fi

(S. A. Gabriel) (a) University of Maryland, College Park, Maryland 20742, USA, (b) Norwegian University of Science and Technology, Trondheim, Norway, (c) Aalto University, Espoo, Finland
Email address: sgabriel@umd.edu