Exact Two-Step Benders Decomposition for Two-Stage Stochastic Mixed-Integer Programs

DOI:
10.48550/arXiv.2306.02849

Document status and date:
Published: 05/06/2023

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Submitted to *Operations Research*
manuscript none

Exact Two-Step Benders Decomposition for Two-Stage Stochastic Mixed-Integer Programs

Şifa Çelik[a], Layla Martin[a,b], Albert H. Schrotenboer[a,b], Tom Van Woensel[a,b]
[a]School of Industrial Engineering, Eindhoven University of Technology, 5612AZ Eindhoven, The Netherlands
[b]Eindhoven AI Systems Institute, Eindhoven University of Technology, 5612AZ Eindhoven, The Netherlands
s.celik@tue.nl, l.martin@tue.nl, a.h.schrotenboer@tue.nl, t.v.woensel@tue.nl

Many real-life optimization problems belong to the class of two-stage stochastic mixed-integer programming problems with continuous recourse. This paper introduces Two-Step Benders Decomposition with Scenario Clustering (TBDS) as a general exact solution methodology for solving such stochastic programs to optimality. The method combines and generalizes Benders dual decomposition, partial Benders decomposition, and Scenario Clustering techniques and does so within a novel two-step decomposition along the binary and continuous first-stage decisions. We use TBDS to provide the first exact solutions for the so-called Time Window Assignment Traveling Salesperson problem. This is a canonical optimization problem for service-oriented vehicle routing; it considers jointly assigning time windows to customers and routing a vehicle among them while travel times are stochastic. Extensive experiments show that TBDS is superior to state-of-the-art approaches in the literature. It solves instances with up to 25 customers to optimality. It provides better lower and upper bounds that lead to faster convergence than related methods. For example, Benders dual decomposition cannot solve instances of 10 customers to optimality. We use TBDS to analyze the structure of the optimal solutions. By increasing routing costs only slightly, customer service can be improved tremendously, driven by smartly alternating between high- and low-variance travel arcs to reduce the impact of delay propagation throughout the executed vehicle route.

Key words: Partial Benders Decomposition, Benders Dual Decomposition, Time Window Assignment, Vehicle Routing, Stochastic Programming
History:

1. Introduction

Two-stage stochastic programming has emerged as a prominent strategy for making decisions in the face of uncertainty. This strategy involves initial first-stage decisions before uncertainty is resolved and second-stage recourse decisions after uncertainty is realized. The objective is to minimize the expected cost associated with both sets of decisions. Numerous real-world applications can be effectively modeled using two-stage stochastic programming, incorporating binary *and* continuous first-stage decisions. For example, Facility Location (Schiffer et al. 2019), Stochastic Inventory Routing (Cui et al. 2023), and Time Window Assignment Vehicle Routing (Vareias, Repoussis, and Tarantilis

2019) are commonly addressed using two-stage stochastic programming. In Facility Location, binary variables determine which locations to open and what customers to assign, while continuous variables define facility capacity. In Stochastic Inventory Routing, binary variables define vehicle routes, while continuous variables represent the delivery amount. Similarly, in Time Window Assignment Vehicle Routing, binary variables determine vehicle routes, and continuous variables model decisions regarding time window assignments to customers. In two-stage stochastic programs, uncertainty is typically represented by constructing a well-defined set of scenarios. Benders decomposition-inspired approaches are commonly utilized to decompose the problem across these scenarios. However, despite employing such techniques, for many problems, it remains challenging to achieve optimal solutions.

Recently, two distinct research streams made notable contributions toward solving general two-stage stochastic programs. The first stream centers around Benders decomposition, which experienced a resurgence in scientific attention with the introduction of Benders dual decomposition (Rahmaniani et al. 2020) and partial Benders decomposition (Crainic et al. 2021). However, when applied to two-stage stochastic programs with binary *and* continuous first-stage decision variables, these approaches often require numerous iterations of generating relatively weak optimality and feasibility cuts, primarily due to low-quality solutions in early iterations (Rahmaniani et al. 2017). The second stream focuses on generating a compact set of scenarios that accurately captures the underlying uncertainty by utilizing scenario clustering techniques aimed at enhancing computational performance in general (see, e.g. Keutchayan, Munger, and Gendreau 2020). To this date, it remains an open question of how such techniques can improve the computational efficiency of state-of-the-art Benders decomposition approaches, such as Benders dual decomposition or partial Benders decomposition.

This paper introduces *Two-Step Benders Decomposition with Scenario Clustering (TBDS)*, an exact method specifically designed to solve two-stage stochastic programs involving binary and continuous first-stage decisions. The effectiveness of TBDS stems from combining two fundamental ideas. Firstly, we employ a novel two-step decomposition approach that effectively handles the first-stage binary and continuous decision variables to generate optimality and feasibility cuts. This two-step decomposition approach enhances the quality of the first-stage continuous variables, resulting in stronger cuts than existing methods. Secondly, TBDS integrates recent advancements in scenario-clustering techniques for stochastic programming (see, e.g., Keutchayan, Ortmann, and Rei 2023), hereby generalizing the principles of partial Benders decomposition. Furthermore, we incorporate the concepts from Benders dual decomposition within TBDS and embed it within a branch-and-cut framework.

The first key concept of TBDS involves decomposing the two-stage stochastic program into a master problem and $N+1$ subproblems, where N represents the number of scenarios. The initial subproblem corresponds to the first step of the two-step decomposition, focusing solely on the binary decision variables from the master problem and considering a single continuous subproblem. This first step offers a notable advantage by producing a more robust solution for the continuous first-stage decisions, which is then utilized in combination with the binary first-stage solution in the subsequent N single-scenario subproblems. Consequently, this two-step approach generates significantly stronger optimality cuts leading to faster convergence to the optimal solution.

The second key concept of TBDS involves generalizing partial Benders decomposition by incorporating *representative* scenarios into the master program. These representative scenarios are carefully selected, allowing us to optimize our decisions in the first stage before the uncertainty is observed. The selection of the scenario set involves a trade-off: a larger set of scenarios reflects underlying uncertainty better but increases computational complexity. There is growing interest in scenario generation and reduction methods to address this trade-off. These methods can be categorized as either distribution-driven, such as those proposed by, e.g., Kleywegt, Shapiro, and Homem-de Mello (2002), Henrion, Küchler, and Römisch (2009), Pflug and Pichler (2015), or problem-driven, as discussed by, e.g., Henrion and Römisch (2018), Keutchayan, Ortmann, and Rei (2023), and this paper.

The main methodological contributions of this paper are threefold:

1. We introduce a new exact solution approach, *Two-Step Benders decomposition with Scenario Clustering (TBDS)*, for solving two-stage stochastic mixed-integer programs with continuous recourse. Our TBDS method incorporates two main ideas: a two-step decomposition focusing on the first-stage binary and continuous variables and the utilization of scenario clustering techniques to generalize partial Benders decomposition.

2. Our TBDS method combines and extends the state-of-the-art approaches in Benders decomposition by incorporating Benders dual decomposition throughout its components. A special case of our TBDS method combines Benders dual decomposition and partial Benders decomposition.

3. We furthermore generalize partial Benders decomposition by incorporating the ideas from Keutchayan, Ortmann, and Rei (2023) to determine representative scenarios of the underlying uncertainty, a problem-driven scenario generation method. As far as the authors know, TBDS is the first exact method that combines problem-driven scenario generation methods with state-of-the-art Benders decomposition.

We evaluate TBDS's performance using the Time Window Assignment Traveling Salesperson Problem with Stochastic Travel Times (TWATSP-ST), a canonical optimization problem in service-oriented vehicle routing. This problem represents a two-stage stochastic program with binary (routing) and continuous (time window assignment) first-stage decision variables and continuous second-stage decision variables related to time window violation. The TWATSP-ST thereby aligns with the active research stream on vehicle routing that considers time window assignment as an integral part of the optimization problem (see, e.g., Spliet, Dabia, and Van Woensel (2018); Vareias, Repoussis, and Tarantilis (2019)), rather than solely adhering to exogenously given time window constraints (see, e.g., Paradiso et al. (2020); Wölck and Meisel (2022)). We summarize our contributions on the intersection of time window assignment and routing as follows:

4. We present the first exact method for jointly optimizing time window assignment and routing by applying TBDS to the TWATSP-ST. Extensive computational experiments demonstrate the superior performance of TBDS on the TWATSP-ST, surpassing existing state-of-the-art approaches like Benders dual decomposition and partial Benders decomposition. TBDS solves instances with up to 25 customers to optimality.

5. By solving the TWATSP-ST to optimality, we automatically cater for delay propagation throughout the execution of a vehicle route. This has either not been accounted for (Vareias, Repoussis, and Tarantilis 2019), or has only be studied with exogenously given time windows (Ehmke, Campbell, and Urban 2015).

6. Our analysis of optimal solutions provides valuable managerial insights, highlighting the advantages of flexible time window assignments and the importance of considering stochasticity in decision-making. Specifically, we compare the performance of the stochastic solution obtained by TBDS against the TSP solution (assuming a vehicle drives the shortest route), expected value solution (assuming travel times follow their expectation), and fixed time window assignment solution (assuming all time windows among customers are of the same length). We show that simultaneously optimizing time windows and routing leads to a noteworthy 12.8% improvement in total costs while incurring only a minor increase in routing costs. Furthermore, we show that the value of the stochastic solution is 6.2%, further highlighting the potential benefits of incorporating stochasticity in decision-making processes.

The remainder of the paper is organized as follows. In Section 2, we briefly overview the relevant background on Benders decomposition. Section 3 presents the TBDS methodology, explaining the associated mathematical models and cuts. Section 4 discusses the application of TBDS on the TWATSP-ST. Section 5 provides computational results on the performance of TBDS and managerial results associated with solving the TWATSP-ST. We conclude this paper and provide avenues for future research in Section 6.

2. Background on Benders Decomposition

In this section, we first provide an overview of Benders decomposition and then discuss Benders dual decomposition, highlighting the fundamental formulations necessary for introducing TBDS in Section 3.

2.1. Benders Decomposition

Dantzig (1955) and Slyke and Wets (1969) were the first to propose algorithms for solving two-stage stochastic linear programs. Slyke and Wets (1969) developed the L-shaped method that exploits the block-angular structure of two-stage stochastic models aligning with the decomposition structure and cut generation principles of Benders decomposition (Benders 1962). Since then, many variants and tailored Benders decomposition algorithms have been proposed, but the core ideas remain similar. Due to the independence of scenarios (reflecting the realization of random vectors), two-stage stochastic programs are decomposable over scenarios, resulting in a master program and one or multiple subproblems, each representing the second-stage problems associated with a scenario $\omega \in \Omega$, where Ω is a finite set of scenarios or realized random events. Throughout this paper, we denote dependency on ω via a subscript. At some points (see Section 4), this notation becomes inconvenient. In that case, we denote it as a functional form.

We introduce Benders decomposition through a two-stage stochastic programming problem of the following generic form

$$\min\{c^T x + \sum_{\omega \in \Omega} p_\omega Q(x,\omega) : Ax = a,\ x \in \mathbb{Z}_+^{n_1}\}, \tag{1}$$

with

$$Q(x,\omega) = \min\{f_\omega^T z_\omega : W_\omega z_\omega = h_\omega - T_\omega x,\ z_\omega \in \mathbb{R}_+^m\} \qquad \forall\ \omega \in \Omega, \tag{2}$$

where $c \in \mathbb{R}^{n_1}$, $A \in \mathbb{R}^{k_1 \times n_1}$, $a \in \mathbb{R}^{k_1}$, $f_\omega \in \mathbb{R}^m$, $W_\omega \in \mathbb{R}^{\ell \times m}$, $h_\omega \in \mathbb{R}^\ell$, and $T_\omega \in \mathbb{R}^{\ell \times n_1}$. Here, $p_\omega \in \mathbb{R}$ denotes the probability of observing scenario $\omega \in \Omega$.

We define the master program (MP) as

$$MP = \min\{c^T x + \theta\} : Ax = a,\ x \in \mathbb{Z}_+^{n_1}, \theta \in \mathbb{R}\}.$$

An auxiliary decision variable θ approximates the recourse function $Q(x) = \sum_{\omega \in \Omega} p_\omega Q(x,\omega)$ in the master program. By utilizing the dual of the Second-Stage Subproblem (2) for a scenario $\omega \in \Omega$, we produce a set of valid inequalities referred to as the *optimality* and *feasibility cuts* for the first-stage decision variables. If the dual of Subproblem (2) for a scenario $\omega \in \Omega$ given solution x^* is unbounded, implying that the primal subproblem is infeasible, we generate feasibility cuts with the unbounded extreme ray to cut off the solution x^*. If the dual of Subproblem (2) for a scenario $\omega \in \Omega$ given solution x^* is feasible, we generate optimality cuts with the extreme points of the dual problem. Optimality cuts provide a lower bound on the expected cost of the recourse function $Q(x,\omega)$.

2.2. Benders Dual Decomposition

Recently, Rahmaniani et al. (2020) introduced Benders dual decomposition (BDD) as a version of Benders decomposition that uses strengthened Benders and Lagrangian cuts.

The BDD method generates Lagrangian cuts by heuristically solving a Lagrangian cut generation problem. Rahmaniani et al. (2020) numerically show that Lagrangian cuts close the gap at the root node substantially for a variety of stochastic *integer* problems. They propose a family of strengthened optimality and feasibility cuts that dominate the classical Benders cuts at *fractional points* of the master problem.

By defining x^* as the current master problem solution, we can reformulate the recourse function $Q(x,\omega)$ for each scenario $\omega \in \Omega$ as

$$Q(x^*,\omega) = \min_{x_\omega, z_\omega} \{ f_\omega^T z_\omega : A x_\omega = a, W_\omega z_\omega = h_\omega - T_\omega x_\omega, \ x_\omega = x^*, \ x_\omega \in \mathbb{R}_+^{n_1}, z_\omega \in \mathbb{R}_+^m \}. \tag{3}$$

The following optimality cut is derived by solving problem (3)

$$\theta \geq \sum_{\omega \in \Omega} p_\omega f_\omega^T \bar{z}_\omega + (x - \bar{x}_\omega)^T \lambda_\omega^*, \tag{4}$$

where $\bar{z}_\omega$ and $\bar{x}_\omega$ represent the optimal solution of the subproblem (3) and λ_ω^* are the dual variables associated with constraints $x_\omega = x^*$.

To strengthen (4), the BDD method prices out the constraints $x_\omega = x^*$ into the objective function using the dual multipliers λ_ω. By doing so, we obtain the following Lagrangian dual problem for each $\omega \in \Omega$

$$\max_{\lambda_\omega} \min_{x_\omega, z_\omega} \{ f_\omega^T z_\omega - \lambda_\omega^T (x_\omega - x^*) : A x_\omega = a, W_\omega z_\omega = h_\omega - T_\omega x \}.$$

Then, given $x^* \in \mathbb{R}^{n_1}$ and $\lambda_\omega^* \in \mathbb{R}^{n_1}$ for $\omega \in \Omega$, let $(\bar{z}_\omega, \bar{x}_\omega)$ be an optimal solution obtained by solving the following problem

$$\min \{ f_\omega^T z_\omega - \lambda_\omega^{*T} (x_\omega - x^*) : A x_\omega = a, W_\omega z_\omega = h_\omega - T_\omega x_\omega, \ x_\omega \in \mathbb{Z}_+^{n_1}, z_\omega \in \mathbb{R}_+^m \}.$$

The *strengthened* optimality cut is valid for MP and given by

$$\theta \geq \sum_{\omega \in \Omega} p_\omega f_\omega^T \bar{z}_\omega + (x - \bar{x}_\omega)^T \lambda_\omega^*. \tag{5}$$

3. Exact Two-Step Benders Decomposition with Scenario Clustering (TBDS)

We propose a general exact solution method for two-stage stochastic programs with mixed binary *and* continuous first-stage decision variables and continuous second-stage decision variables. The overall method is a branch-and-cut approach, of which the algorithmic details are discussed at the end of this section. We assume that our problems are primal feasible and bounded. Let $\xi(\cdot)$ represent a random vector on a finite scenario sample space Ω. Specifically, ξ_ω denotes the particular realization of the random vector in scenario $\omega \in \Omega$, with p_ω as the associated probability. In the remainder of this paper, we work directly with p_ω.

Three elements play a key role in the *Two-Step Benders Decomposition with Scenario Clustering (TBDS)*:

1. We consider a Master Problem (MP) with two associated (sets of) subproblems. The first subproblem, SP1, arises from fixing the binary variables in the MP, leading to a linear programming problem acting on the continuous first-stage variables and all continuous second-stage variables. The second subproblem SP2 is obtained by fixing the binary first-stage variables obtained from the master problem and taking the continuous second-stage variables as the solution of SP1, resulting in one subproblem SP2 per scenario.

2. Considering scenarios and the associated second-stage constraints directly in the MP prevents generating superfluous optimality cuts. It influences the total number of subproblems SP2 from which we generate these optimality cuts. We generalize this idea by considering recent advances in scenario clustering techniques within stochastic programming. This helps to reduce the number of weak first-stage solutions in early iterations of the method.

3. We use Benders dual decomposition to strengthen the optimality cuts we derive from our subproblems, and embed all the aforementioned concepts in a branch-and-cut algorithm.

In the remainder of this paper, we consider the following two-stage stochastic mixed-integer program with relatively complete continuous recourse

$$\min_{x,y,z} \quad c^T x + d^T y + \sum_{\omega \in \Omega} p_\omega f_\omega^T z_\omega \tag{6}$$

$$\text{s.t.} \quad W_\omega x + T_\omega y + S_\omega z_\omega \geq h_\omega \qquad \forall\, \omega \in \Omega, \tag{7}$$

$$x \in \mathcal{X}, \; y \in \mathcal{Y}, \tag{8}$$

$$z_\omega \in \mathbb{R}^m_+ \qquad \forall\, \omega \in \Omega. \tag{9}$$

where $c \in \mathbb{R}^{n_1},\ d \in \mathbb{R}^{n_2},\ f_\omega \in \mathbb{R}^m,\ W_\omega \in \mathbb{R}^{\ell \times n_1}, T_\omega \in \mathbb{R}^{\ell \times n_2},\ S_\omega \in \mathbb{R}^{\ell \times m}$, and $h_\omega \in \mathbb{R}^\ell$. We denote first-stage constraints and their domains compactly as $x \in \mathcal{X} := \{Ax = a, x \in \mathbb{Z}^{n_1}_+\}$ and $y \in \mathcal{Y} := \{Bx = b, y \in \mathbb{R}^{n_2}_+\}$ where $A \in \mathbb{R}^{k_1 \times n_1},\ a \in \mathbb{R}^{k_1},\ B \in \mathbb{R}^{k_2 \times n_2}$, and $b \in \mathbb{R}^{k_2}$. Note the first-stage decision variables are binary variables x and continuous variables y.

We reformulate the two-stage stochastic mixed-integer problem with continuous recourse by explicitly considering Ω_{MP} (not necessarily subset of Ω) in the master problem (via second-stage constraints

(7)) and $\Omega_{SP} := \Omega \setminus \Omega_{MP}$. We detail the construction of Ω_{MP} in Section 3.2. The scenarios Ω_{SP} impose $|\Omega_{SP}|$ subproblems SP2, from which we derive optimality cuts for our novel two-step decomposition.

Before we detail TBDS throughout the remainder of this section, we already state the Master Problem (MP)

$$
\begin{aligned}
MP = \min_{x,y} \quad & c^T x + \Theta + \sum_{\omega \in \Omega^1_{MP}} p_\omega f_\omega^T z_\omega && && (10)\\
\text{s.t.} \quad & W_\omega x + T_\omega y + S_\omega z_\omega \geq h_\omega && \forall\, \omega \in \Omega_{MP}, && (11)\\
& \Theta \geq d^T y + \sum_{\omega \in \Omega_{SP}} p_\omega \theta_\omega, && && (12)\\
& \Theta \geq \Big(d^T \bar{y} + \sum_{\omega \in \Omega_{SP}} p_\omega f_\omega^T \bar{z}_\omega + (x - \bar{x})\lambda^* \Big)_i && \forall\, i \in I, && (13)\\
& \theta_\omega \geq \Big(p_\omega f_\omega^T \hat{z}_\omega + (x - \hat{x}_\omega)\nu_\omega^* + (y - \hat{y}_\omega)\eta_\omega^* \Big)_j && \forall\, j \in J_\omega, \omega \in \Omega_{SP}, && (14)\\
& 0 \ \geq \big(\mathbb{1}^T \epsilon + (x - \bar{x})\lambda^* \big)_k && \forall\, k \in K, && (15)\\
& x \in \mathcal{X}, y \in \mathcal{Y}, \Theta \in \mathbb{R}_+, && && (16)\\
& z_\omega \in \mathbb{R}^m_+, \theta_\omega \in \mathbb{R}_+ && \forall\, \omega \in \Omega. && (17)
\end{aligned}
$$

We introduce the auxiliary variables Θ and $\theta_\omega, \forall\, \omega \in \Omega_{SP}$. Here, Θ approximates the objective function of SP1 while θ_ω approximates the SP2 subproblem cost. In MP, we minimize the total cost associated with x, θ, and the expected second-stage cost of included scenarios (Ω^1_{MP}). Constraints (11) are the second-stage constraints. Constraint (12) is called the *subproblem connectivity constraint* and its use will be explained later in this section. Constraints (13) - (14) are the *optimality cuts*. Constraints (15) are the *feasibility cuts*. The sets I, J_ω and K refer to the complete set of feasibility and/or optimality cuts. The parameters and variables appearing in these constraints will be detailed in the remainder of this section at the appropriate moments to enhance readability. Constraints (16) and (17) restrict the variable domains.

3.1. Subproblem Decomposition, Optimality Cuts, and Subproblem-Connectivity Constraint

Classic Benders decomposition decomposes first- and second-stage decisions, but the continuous first-stage variables reduce the efficiency of the resulting optimality cuts as MP loses the information regarding the second stage. Therefore, TBDS proposes a two-step decomposition. After solving MP, we first take the binary variables fixed and derive optimality and feasibility cuts based on SP1. Then, we take the solution of the continuous first-stage variables from SP1 together with the binary variables from MP and derive optimality cuts on the scenario subproblems SP2. We will detail each of these cuts consecutively.

The solution values of the MP define lower bounds for (6) - (9). We solve the MP using branch-and-cut and thus dynamically include the aforementioned cuts while exploring the branch-and-bound tree. At each node of the branch-and-bound tree, the solution x^* is fixed in the primal subproblem (SP1)

$$SP1(x^*) = \min_{y,z}\{d^T y + \sum_{\omega\in\Omega_{SP}} p_\omega f_\omega^T z_\omega \mid W_\omega x + T_\omega y + S_\omega z_\omega \geq h_\omega \ \forall\ \omega\in\Omega,\ x = x^*, y\in\mathcal{Y}, z_\omega\in\mathbb{R}_+^m\}.$$

Let λ^* be the dual multipliers associated with the constraints $x = x^*$. If primal subproblem (SP1) is infeasible for x^*, we solve the feasibility problem

$$\min_{x,y,z,\epsilon}\{\mathbb{1}^T\epsilon : W_\omega x + T_\omega y + S_\omega z_\omega + \bar{\epsilon} \geq h_\omega \ \forall\ \omega\in\Omega_{SP},\ x = x^*, y\in\mathcal{Y}, z_\omega\in\mathbb{R}_+^m, \epsilon\in\mathbb{R}_+^\ell\}. \tag{18}$$

This generates a feasibility cut of the form

$$0 \geq \mathbb{1}^T\bar{\epsilon} + (x-\bar{x})^T\lambda^*, \qquad \textit{(Feasibility Cut)} \tag{19}$$

where $\bar{\epsilon}$ and $\bar{x}$ refer to the values of ϵ and x in the optimal solution to the feasibility problem (18), and $\mathbb{1}$ is a vector of ones of size ℓ.

If the primal subproblem (SP1) returns a feasible solution $(\bar{y}, \bar{z})$, we derive the optimality cut

$$\Theta \geq d^T\bar{y} + \sum_{\omega\in\Omega_{SP}} p_\omega f_\omega^T \bar{z}_\omega + (x-\bar{x})^T\lambda^* \qquad \textit{(Aggregated Optimality Cut)} \tag{20}$$

We do not decompose SP1 over the scenarios because it is a linear program and, thus, easy to solve computationally. Including all scenarios provides more information and thus results in relatively 'good' first-stage decisions. The second (set of) subproblems is obtained by decomposing into single scenario second-stage subproblems for a given $\bar{y}$ obtained as the optimal solution of SP1 and x^* as the optimal solution to (the linear relaxation of) MP. That is,

$$SP2(x^*,\bar{y},\omega) = \min_{x_\omega,y_\omega,z_\omega}\{f_\omega^T z_\omega : W_\omega x_\omega + T_\omega y + S_\omega z_\omega \geq h_\omega, x_\omega = x^*, y_\omega = \bar{y},\ x_\omega\in\mathbb{R}_+^{n_1}, y_\omega\in\mathbb{R}_+^{n_2}, z_\omega\in\mathbb{R}_+^m\}.$$

Note we only construct SP2 if SP1 is feasible; thus, SP2 is feasible by construction. Let ν_ω and η_ω denote the dual multipliers to the constraints $y_\omega = \bar{y}$ and $x_\omega = x^*$ in $SP2(x^*,\bar{y},\omega)$, respectively. Then, the following optimality cut can be derived

$$\theta_\omega \geq p_\omega f_\omega^T\hat{z}_\omega + (x-\hat{x}_\omega)^T\nu_\omega^* + (y-\hat{y}_\omega)^T\eta_\omega^* \qquad \textit{(Scenario Optimality Cut)}, \tag{21}$$

where $(\hat{x}_\omega, \hat{y}_\omega, \hat{z}_\omega)$ is the optimal solution of $SP2(x^*,\bar{y},\omega)$.

If x^* is fractional and feasible, the steps presented in Section 2.2 improve the optimality cut (21). We price out the constraints $x_\omega = x^*$ into the objective function of SP2 with dual multiplier ν_ω. The resulting subproblem then asks for solving

$$\min_{x_\omega,y_\omega,z_\omega}\{f_\omega^T z_\omega + (x_\omega - x^*)^T\nu_\omega^* : W_\omega x_\omega + T_\omega y + S_\omega z_\omega \geq h_\omega, y_\omega = \bar{y},\ x_\omega\in\mathbb{Z}_+^{n_1}, y_\omega\in\mathbb{R}_+^{n_2}, z_\omega\in\mathbb{R}_+^m\} \tag{22}$$

Then, given ν_ω^* for $\omega \in \Omega$, let $(\tilde{y}_\omega, \tilde{x}_\omega, \tilde{z_\omega})$ be an optimal solution obtained by solving problem (22). The strengthened optimality cut

$$\theta_\omega \geq p_\omega f_\omega^T \tilde{z}_\omega + (x - \tilde{x}_\omega)^T \nu_\omega^* + (y - \tilde{y}_\omega)^T \eta_\omega^*, \tag{23}$$

where η_ω^* is the dual variable associated with the constraint $y_\omega = \bar{y}$ in (22) is valid for MP.

We now derived two sets of optimality cuts, one for the aggregated subproblem SP1 and $|\Omega_{SP}|$ for each individual scenario subproblem, along the lines of our two-step decomposition. To obtain a valid MP formulation, recall the *Subproblem-Connectivity Constraint* $\Theta \geq d^T y + \sum_{\omega \in \Omega} p_\omega \theta_\omega$ in MP. This ensures that we also include θ_ω for $\omega \in \Omega$ via Θ in the objective function of the MP such that constraints (21) and (23) can improve the lower bound in each iteration.

Proposition 1. *For a given feasible solution $(\hat{x}_\omega, \hat{y}_\omega, \hat{z}_\omega)$ of SP2, Constraint* (21) *is a valid optimality cut for MP if*

$$\Theta \geq d^T y + \sum_{\omega \in \Omega_{SP}} p_\omega \theta_\omega \tag{24}$$

is part of MP.

Proof: Consider the MP' below.

$$\begin{aligned}
MP' = \min_{x,y} \quad & c^T x + d^T y + \sum_{\omega \in \Omega_{SP}} p_\omega \theta_\omega + \sum_{\omega \in \Omega_{MP}^1} p_\omega f_\omega^T z_\omega \\
\text{s.t.} \quad & W_\omega x + T_\omega y + S_\omega z_\omega \geq h_\omega && \forall\, \omega \in \Omega_{MP}, \\
& \theta_\omega \geq \left(p_\omega f_\omega^T \hat{z}_\omega + (x - \hat{x}_\omega)\nu_\omega^* + (y - \hat{y}_\omega)\eta_\omega^* \right)_j && \forall\, j \in J_\omega, \omega \in \Omega_{SP}, \\
& 0 \ \geq \left(\mathbb{1}^T \epsilon + (x - \bar{x})\lambda^* \right)_k && \forall\, k \in K, \\
& x \in \mathcal{X},\ y \in \mathcal{Y},\ z_\omega \in \mathbb{R}_+^m,\ \Theta \in \mathbb{R}^+,\ \theta_\omega \in \mathbb{R}^+ && \forall\, \omega \in \Omega.
\end{aligned}$$

We want to show that the Constraint (21) is a valid optimality cut for the MP with the inclusion of Constraint (24) by showing the equivalence of MP' and MP. MP' yields the same objective value as in MP when Constraint (21) is added iteratively (the proof follows from the L-shaped method, see Van Slyke and Wets (1969)). Keeping in mind that the above problem is a minimization problem, Θ is lower bounded by $d^T y + \sum_{\omega \in \Omega_{SP}} p_\omega \theta_\omega$ and constraint (20), i.e.,

$$\Theta \geq \max \left\{ d^T y + \sum_{\omega \in \Omega_{SP}} p_\omega \theta_\omega, \quad d^T \bar{y} + \sum_{\omega \in \Omega_{SP}} p_\omega f_\omega^T \bar{z_\omega} + (x - \bar{x})^T \lambda^* \right\}$$

We can linearize this constraint and rewrite the problem equivalently as

$$MP' = \min_{x,y} \quad c^T x + \Theta + \sum_{\omega \in \Omega_{MP}^1} p_\omega f_\omega^T z_\omega$$

$$
\begin{aligned}
\text{s.t.}\quad & W_\omega x + T_\omega y + S_\omega z_\omega \geq h_\omega && \forall\, \omega \in \Omega_{MP},\\
& \Theta \geq d^T y + \sum_{\omega \in \Omega_{SP}} p_\omega \theta_\omega, && \\
& \Theta \geq d^T \bar{y} + \sum_{\omega \in \Omega_{SP}} p_\omega f_\omega^T \bar{z_\omega} + (x-\bar{x})^T \lambda^*, && \\
& \theta_\omega \geq \left(p_\omega f_\omega^T \hat{z}_\omega + (x-\hat{x}_\omega)\nu_\omega^* + (y-\hat{y}_\omega)\eta_\omega^*\right)_j && \forall\, j \in J_\omega, \omega \in \Omega_{SP},\\
& 0\ \geq \left(\mathbb{1}^T \epsilon + (x-\bar{x})\lambda^*\right)_k && \forall\, k \in K,\\
& x \in \mathcal{X},\ y \in \mathcal{Y},\ z_\omega \in \mathbb{R}_+^m,\ \theta_\omega \in \mathbb{R}^+ && \forall\, \omega \in \Omega.
\end{aligned}
$$

which yields the exactly the same MP as in (10) - (17). □

We summarize the main contribution of our TBDS method, from a Benders decomposition perspective, in Theorem 1.

THEOREM 1. *Let Z_{MIP} and Z_{MP}^{RN} be the optimal objective value of problem* (6) - (9) *and the optimal objective value of MP with a finite number of proposed optimality and feasibility cuts, respectively, then, $Z_{MIP} = Z_{MP}^{RN}$.*

Proof: The proof follows from Van Slyke and Wets (1969) and Proposition 1. Let $Z_{MP}^{RN\prime}$ be the optimal objective value of of MP' with a finite number of proposed optimality and feasibility cuts. By Van Slyke and Wets (1969), we know $Z_{MIP} = Z_{MP}^{RN\prime}$. It is shown that (with Proposition 1) $Z_{MP}^{RN\prime} = Z_{MP}^{RN}$. Hence, we can conclude $Z_{MIP} = Z_{MP}^{RN}$. □

3.2. Design of the scenario set Ω_{MP}

Relatively complete continuous recourse in stochastic mixed-integer programs typically entails weak bounds and many (superfluous) iterations of generating cuts as the MP loses all the information with the second stage variables (Rahmaniani et al. 2017). To overcome this issue, we adopt and generalize the idea of partial Benders decomposition (Crainic et al. 2021). In line with partial Benders decomposition, we include second-stage constraints associated with a subset of scenarios Ω_{MP}, via constraints (11). However, partial Benders decomposition designs the set Ω_{MP} using a *row covering strategy* to eliminate many feasibility cuts. Instead, we determine a 'representative' scenario subset Ω_{MP}.

The representative scenarios in the master problem $\Omega_{MP} = \Omega_{MP}^1 \cup \Omega_{MP}^2$ comprise 'actual' scenarios $\Omega_{MP}^1 \subseteq \Omega$ and 'artificial' scenarios Ω_{MP}^2. Let Ω_{MP}^2 be a set of artificial scenarios created by convex combinations of scenarios in Ω_{MP}. We first detail the Ω_{MP}^1 selection. We cluster the scenarios following Definition 1, and select the representative scenarios of each cluster for Ω_{MP}^1.

DEFINITION 1. Let K be the number of clusters. The set $\Omega^1_{MP} = \{r_1, \ldots, r_K\}$ is constructed as follows:

Step 1. Compute the opportunity-cost matrix $\mathbb{V} = (V_{ij})_{|\Omega|\times|\Omega|}$ where

$$V_{ij} = SP2((\hat{x}_i, \hat{y}_i), \xi(\omega_j)) \qquad \forall (i,j) \in \Omega$$

where $(\hat{x}_i, \hat{y}_i)$ is the optimal solution of the one-scenario subproblem:

$$(\hat{x}_i, \hat{y}_i) \in \arg\min_{x,y} SP2(x, y, \xi(\omega_i)) \qquad \forall i \in \Omega$$

Step 2. Find a partition of the set Ω into K clusters $C_1, \ldots, C_K$ and their representative scenarios $r_1 \in C_1, \ldots, r_K \in C_K$ such that

$$r_k = \arg\min \left| V_{r_k, r_k} - \frac{1}{|C_k|} \sum_{j \in C_k} V_{r_k, j} \right| \tag{25}$$

By minimizing the clustering error (equation (25)), we create clusters that best fit the average cost function of all clusters. Appendix A gives an equivalent mixed-integer program to solve equation (25). The reader is referred to Keutchayan, Ortmann, and Rei (2023) for more insight.

The above procedure adds K so-called representative scenarios into MP by means of constraints (11). This reduces the root node optimality gap by improving the linear relaxation of the master problem. The value of K should be carefully selected, however. A too large K leads to overpopulating the master problem, increasing the solution time. We provide a detailed analysis of the value of K in Section 5.

Additionally, we construct artificial scenarios Ω^2_{MP} based on scenarios in Ω. Adding artificial scenarios into the master program influences the first-stage variables, improving the lower bound. To generate artificial scenarios $\bar{\omega} \in \Omega^2_{MP}$, we use convex combinations of scenarios in Ω and add the constraints (11) to MP.

DEFINITION 2. Let $\alpha^{\bar{\omega}}_{\omega} \geq 0$, for $\omega \in \Omega$ such that $\sum_{\omega \in \Omega} \alpha^{\bar{\omega}}_{\omega} = 1$. Then, the realization of random vector for artificially generated scenario $\bar{\omega} \in \Omega^2_{MP}$ is defined as

$$W_{\bar{\omega}} = \sum_{\omega \in \Omega} \alpha^{\bar{\omega}}_{\omega} W_{\omega}, \quad T_{\bar{\omega}} = \sum_{\omega \in \Omega} \alpha^{\bar{\omega}}_{\omega} T_{\omega}, \quad S_{\bar{\omega}} = \sum_{\omega \in \Omega} \alpha^{\bar{\omega}}_{\omega} S_{\omega}.$$

We can guarantee the same objective value of the problem (6) - (9) with the inclusion of artificial scenarios by defining the second stage decision variables for an artificial scenario as $y_{\bar{\omega}} = \sum_{\omega \in \Omega} \alpha^{\bar{\omega}}_{\omega} y_{\omega}$. Convex combinations of scenarios, as suggested by Crainic et al. (2021), can create dominance relationships among scenarios, resulting in fewer feasibility cuts and a considerable reduction in the optimality gap upon termination. This technique also improves the number of instances that can be solved optimally within the time limit. Overall, including a set of scenarios Ω_{MP} in the master problem can strengthen it and lead to faster convergence.

Algorithm 1 Two-Step Benders Decomposition within Branch and Cut Method

1: **Input:** $SP1(\cdot)$,$\{SP2(\cdot,\omega)\}_{\omega\in\Omega_{SP}}$
2: **Output:** (x^*,y^*,Θ)
3: Initialize $\Theta=-\infty$, $\theta_\omega=-\infty \qquad \forall\omega\in\Omega_{SP}$,
4: Start a callback procedure
5: Callback:
6: **if** $x^*\notin\mathcal{X}$ **then**
7:   **if** $SP1(x^*)$ is a finite number **then**
8:     Generate aggregated scenario optimality cut (20), add to the MP
9:     Solve SP2 and generate strengthened scenario optimality cut (23), add to the MP
10:   **else**
11:     Generate the feasibility cut (19)
12:   **end if**
13: **else**
14:   Solve $SP1(x^*)$ and get the solution y^*
15:   **if** $\Theta^*<SP1(x^*)$ **then**
16:     Generate aggregated scenario optimality cut (20), add to the MP
17:     **for** $\omega\in\Omega_{SP}$ **do**
18:       Solve $SP2(x^*,y^*,\omega)$
19:       **if** $\theta^*_\omega<SP2(x^*,y^*,\omega)$ **then**
20:         Generate the scenario optimality cut (21), add to the MP
21:       **end if**
22:     **end for**
23:   **else**
24:     The solution (x^*,y^*) is optimal
25:   **end if**
26: **end if**

3.3. The TBDS Algorithm

We provide an efficient algorithmic implementation of TBDS along the lines of branch-and-cut to obtain an efficient algorithm for solving two-stage stochastic mixed-integer programs with continuous recourse. The TBDS algorithm dynamically adds optimality and feasibility cuts during the branch-and-bound procedure. Note the branch-and-bound procedure ensures integrality of the x variables, so that $x\in\mathcal{X}$.

An algorithmic description of TBDS is provided in Algorithm 1. We distinct two cases for any arbitrary branch-and-bound node. First, if the associated solution x is fractional (line 6), we create a feasibility cut (19) in case SP1 is infeasible (line 11). If the associated solution is feasible (lines 8, 9), however, we generate the strengthened optimality cuts (23) as these cuts are tighter than the optimality cuts (21) for fractional first-stage solutions (Rahmaniani et al. 2020). Second, if the associated solution x is integer, we generate aggregated optimality cuts (20) and scenario optimality cuts (21). Note that strengthening the scenario optimality cuts for integer solutions is not useful since the optimality cuts (23) are not tight as the optimality cuts (21) (Rahmaniani et al. 2020). Due to our relatively complete recourse assumption, we do not need feasibility cuts at integer solutions.

Finally, we like to stress that any other row generation procedure on feasibility on $\mathcal{X}$ and $\mathcal{Y}$ can easily be included in this procedure.

4. Example application of TBDS

We apply TBDS to the Time Window Assignment Traveling Salesperson Problem with Stochastic Travel Times (TWATSP-ST). It concerns the a-priori joint optimization of a vehicle route and the assignment of time windows to the customers, in the presence of travel time uncertainty, initially introduced by Jabali et al. (2015). Recently, Vareias, Repoussis, and Tarantilis (2019) extend the model of Jabali et al. (2015) and develop a two-stage heuristic to solve the problem. To this date, only heuristic approaches appeared in the literature for the TWATSP-ST. We will apply TBDS to the TWATSP-ST to obtain the first exact solutions. In the subsequent parts of this section we provide a two-stage stochastic programming formulations with binary and continuous first-stage variables and continuous second-stage variables for the TWATSP-ST, and show explicit formulations for the optimality and feasibility cuts when applying TBDS to the TWATSP-ST.

The TWATSP-ST is defined on a graph $G = (V, A)$, where $V = \{0, \dots, n\}$ is the set of nodes and $A := \{(i, j) \in V \times V : i \neq j\}$ is the set of arcs. Node 0 acts as the depot at which the vehicle starts its tour and all other nodes represent customers. The vehicle has a shift duration of length T. Each arc $(i, j) \in A$ has a known distance $d_{ij} \geq 0$. Each customer $i \in V^+ := V \setminus \{0\}$ faces a deterministic service time $s_i \geq 0$. We assume travel times over the arcs are stochastic with known distribution. Let $\xi = \{t_{ij}\}_{(i,j)\in A}$ represent the stochastic travel time vector on a scenario sample space Ω. $\xi(\omega) = \{t_{ij}(\omega)\}_{(i,j)\in A}$ denotes the particular realization of the travel time over arc $(i, j) \in A$ in a scenario $\omega \in \Omega$. In this way, we intrinsically cater for delay propagation, unlike Vareias, Repoussis, and Tarantilis (2019).

In this context, the TWATSP-ST makes two inter-dependent, a priori decisions: i) A vehicle route visiting all customers in V^+, starting and ending at the depot, and ii) a time window assignment $[t_i^s, t_i^e]$ for each customer $i \in V^+$. The objective of the TWATSP-ST is to minimize a weighted sum of expected earliness and lateness at the assigned time windows, the width of the assigned time window, the expected shift overtime of the vehicle, and the total distance the vehicle travels.

We encode the routing decision with variables $x_{ij} \in \{0, 1\}$ for all $(i, j) \in A$. Together with the time window assignment variables $t_i^s, t_i^e \geq 0$, these form the first-stage decisions in the TWATSP-ST. After realization of uncertainty, for each scenario $\omega \in \Omega$, we can determine for each customer $i \in V^+$ the departure time $w_i(\omega)$, the earliness $e_i(\omega)$ and the lateness $l_i(\omega)$ relative to the assigned time window $[t_i^s, t_i^e]$, and the shift overtime $o(\omega)$ relative to the shift duration length T.

Then, we formulate the TWATSP-ST as the following two-stage stochastic mixed-integer program with continuous recourse.

$$\min \quad \sum_{i\in V}\sum_{j\in V\setminus\{i\}} d_{ij}x_{ij} + \sum_{j\in V^+}\varphi\left(t_i^e - t_i^s\right) + \sum_{\omega\in\Omega} p(\omega)Q(x,t^s,t^e,\omega) \tag{26}$$

$$\text{s.t.} \quad \sum_{i\in V\setminus\{j\}} x_{ij} = \sum_{i\in V\setminus\{j\}} x_{ji} = 1 \qquad \forall j\in V, \tag{27}$$

$$x_{ii} = 0 \qquad \forall i\in V, \tag{28}$$

$$\sum_{i\in S}\sum_{j\notin S} x_{ij} \geq 1 \qquad S\subseteq V, 1\leq |S|\leq |V^+|, \tag{29}$$

$$t_i^e - t_i^s \geq s_i \qquad \forall i\in V^+, \tag{30}$$

$$x_{ij}\in\{0,1\} \qquad \forall i\in V, j\in V, \tag{31}$$

$$t_i^e, t_i^s \in \mathbb{R}_+ \qquad \forall i\in V^+. \tag{32}$$

Here, $\varphi \geq 0$ is an exogenously set weight factor. Constraints (27) and (28) ensure that the vehicle visits each customer exactly once. Constraints (29) eliminate sub-tours, and constraints (30) ensure that the time window assignment respects the service time at each customer.

The recourse function $Q(x,t^s,t^e,\omega)$, as part of the Objective (26), gives the value of the expected cost of the incurred earliness and lateness cost associated with the time window assignment and the expected shift overtime cost.

$$Q(x,t^s,t^e,\omega) := \min \quad \sum_{j\in V^+}\phi\left(e_j(\omega) + l_j(\omega)\right) + \psi o(\omega) \tag{33}$$

$$\text{s.t.} \quad w_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) + s_j - (1 - x_{ij})M \qquad \forall i\in V, j\in V, \tag{34}$$

$$e_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) - t_i^s - (1 - x_{ij})M \qquad \forall i\in V, j\in V, \tag{35}$$

$$l_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) + s_j - t_i^e - (1 - x_{ij})M \qquad \forall i\in V, j\in V, \tag{36}$$

$$o(\omega) \geq w_i(\omega) + t_{i0}(\omega) - T \qquad \forall i\in V^+, \tag{37}$$

$$w_0(\omega) = t_0, \tag{38}$$

$$w_j(\omega), e_j(\omega), l_j(\omega) \in \mathbb{R}_+ \qquad \forall j\in V^+, \tag{39}$$

$$o(\omega)\in\mathbb{R}_+. \tag{40}$$

Here, ψ and ϕ are weight factors for the earliness and lateness concerning the time window assignment and the overtime of the vehicle, respectively. The objective function of the second-stage problem (33) is a function of first-stage variables (x, t^s, t^s) and a realization (or a scenario) of $\xi(\omega)$. Constraints (34) - (37) determine departure time, earliness, lateness, and overtime for each scenario. We set M in (34) - (36) equal to the longest total travel time among all scenarios. We require tours to start from the depot at a predetermined time t_0 (Constraints (38)). Constraints (31), (32), (39) and (40) define the variable domain.

Model (26) - (40) is a two-stage stochastic mixed-integer program with continuous recourse. We define the master problem of TBDS and the associated cuts in the remainder of this section.

4.1. Master Program

As stated in Section 3, our TBDS method consists of a Master Program (MP) and two (sets) of subproblems SP1 and SP2. We refer the reader to Appendices B and C for the formulation of SP1 and SP2. The MP for the TWATSP-ST is given by:

$$
\begin{aligned}
(MP):\ \min \quad & \sum_{j \in V} \sum_{i \in V \setminus \{j\}} d_{ij} x_{ij} + \Theta + \sum_{\omega \in \Omega^1_{MP}} p(\omega) \left(\sum_{j \in V^+} \phi \left(e_j(\omega) + l_j(\omega)\right) + \psi o(\omega) \right) && (41)\\
\text{s.t.} \quad & \sum_{i \in V \setminus \{j\}} x_{ij} = \sum_{i \in V \setminus \{j\}} x_{ji} = 1 && \forall j \in V, \quad (42)\\
& x_{ii} = 0 && \forall i \in V, \quad (43)\\
& \sum_{i \in S} \sum_{j \notin S} x_{ij} \geq 1 && S \subseteq V, 1 \leq |S| \leq |V^+|, \quad (44)\\
& t^e_i - t^s_i \geq s_i && \forall i \in V^+, \quad (45)\\
& \Theta \geq \sum_{j \in V^+} \varphi\left(t^e_i - t^s_i\right) + \sum_{\omega \in \Omega_{SP}} p(\omega) \theta(\omega), && (46)\\
& \text{Second-stage Constraints } (\Omega_{MP}), && (47)\\
& \left(\text{Aggregated Optimality Cut } (\Omega_{SP})\right)_i, && \forall i \in I \quad (48)\\
& \left(\text{Scenario Optimality Cut } (\Omega_{SP})\right)_j, && \forall j \in J \quad (49)\\
& \left(\text{Feasibility Cut } (\Omega_{SP})\right)_k, && \forall k \in K \quad (50)\\
& x_{ij} \in \{0,1\} && \forall i \in V,\ j \in V, \quad (51)\\
& t^e_i, t^s_i \in \mathbb{R}_+ && \forall i \in V^+, \quad (52)\\
& e_i(\omega), l_i(\omega) \in \mathbb{R}_+ && \forall i \in V^+, \omega \in \Omega^1_{MP}, \quad (53)\\
& o(\omega) \in \mathbb{R}_+ && \forall \omega \in \Omega^1_{MP}, \quad (54)\\
& \Theta \in \mathbb{R}, && (55)\\
& \theta(\omega) \in \mathbb{R} && \forall\ \omega \in \Omega_{SP}. \quad (56)
\end{aligned}
$$

Here Θ (and $\theta(\omega)$) provide a lower bound on the expected second-stage cost and the time window assignment cost. Constraints (42)-(45), (51)-(52) are the first-stage constraints. The scenario set Ω_{SP} in constraint (47) is formed as detailed in Section 3.2. For the scenarios $\omega \in \Omega_{MP}$, second-stage constraints (34)-(40) are included in the MP. Constraint (46) is the Subproblem-Connectivity Constraint. The optimality and feasibility cuts (48)-(50) are made specific in the next subsection.

4.2. Cuts

In line with Section 3, we define a subproblem SP1 taking as input the binary first-stage decisions, i.e., the routing decision in the TWATSP-ST. The second series of subproblems SP2 for each scenario $\omega \in \Omega_{SP}$ are then obtained via the two-step decomposition as outlined in Section 3, i.e., as input we take the binary first-stage decisions and solution of the time window assignment variables after solving SP1. For completeness, we provide the formulation of SP1 for the TWATSP-ST in Appendix B.

Aggregated Optimality Cut. For a given solution x^* of MP that is feasible for SP1, let $(\lambda^{1*}, \lambda^{2*}, \lambda^{3*}, \mu^*, \nu^*, \pi^*)$ indicate the values of dual multipliers in SP1; then,

$$\Theta \geq \sum_{\omega\in\Omega_{SP}} \sum_{i,j\in V} \left(\left(\lambda_{ij}^{1*}(\omega) + \lambda_{ij}^{3*}(\omega) \right) \left(t_{ij}(\omega) + s_j - (1-x_{ij}) M \right) + \lambda_{ij}^{2*}(\omega) \left(t_{ij}(\omega) - (1-x_{ij}) M \right) \right)$$
$$+ \sum_{\omega\in\Omega_{SP}} \sum_{i\in V^+} \left(\pi^*(\omega) t_0 + \mu_i^*(\omega) \left(t_{i0}(\omega) - T \right) \right) + \sum_{i\in V^+} \nu_i^* s_i$$

is the Aggregated Optimality Cut for MP.

For the second step of our decomposition, we formulate SP2 as detailed in Section 3.1. It takes the routing decisions from MP and time window assignments from SP1 as input. Recall SP2 is given in Appendix C. Via the auxiliary decision variable $\theta(\omega)$ we approximate the cost function of each scenario $\omega \in \Omega_{SP}$ (or the objective function value of each SP2).

Feasibility Cuts. If the solution x^* of MP is infeasible for SP1, we generate the feasibility cut

$$0 \geq \mathbb{1}^T \bar{\epsilon} + (x - \bar{z})^T \lambda^*$$

where $\bar{\epsilon}$ and $\bar{z}$ are the optimal values of the ϵ and z variables in the feasibility problem in Appendix D. λ^* is the value of the associated dual variable.

Scenario Optimality Cuts. For a given feasible solution $(x^*,\ t^{s*},\ t^{e*})$, for $\omega \in \Omega_{SP}$, let $(\bar{z}, \bar{q}^s, \bar{q}^e, \bar{e}, \bar{l}, \bar{o})$ be the optimal solution of SP2 and $(\beta,\ \lambda,\ \eta)$ indicate the values of the dual multipliers related to first-stage variables; then,

$$\theta(\omega) \geq \sum_{j\in V^+} \left(\phi \left(\bar{e_j}(\omega) + \bar{l_j}(\omega) \right) + \psi \bar{o}(\omega) + \lambda_j (t_i^s - \bar{q}_j^s) + \eta_j (t_j^e - \bar{q}_j^e) \right) + \sum_{i\in V} \sum_{j\in V} \beta_{ij} (x_{ij} - \bar{z}_{ij}) \quad (57)$$

is the Scenario Optimality Cut for $\omega \in \Omega_{SP}$ for MP.

Strengthened Scenario Optimality Cuts. For a feasible fractional MP solution $(x^*,\ t^{s*},\ t^{e*})$, we update the cut (57) by acquiring the following Lagrangian dual problem of SP2 with the objective function

$$\max_{\beta,\lambda,\eta} \min \sum_{i\in V} \sum_{j\in V} \beta (x_{ij}^* - z_{ij}) + \lambda_i (t_i^{s*} - q_i^s) + \eta_i (t_i^{e*} - q_i^e) + \sum_{j\in V^+} \phi \left(e_j(\omega) + l_j(\omega) \right) + \psi o(\omega).$$

Given (x^*, t^{s*}, t^{e*}) and $(\beta,\ \lambda,\ \eta)$, let $(\bar{z}, \bar{q}^s, \bar{q}^e, \bar{e}, \bar{l}, \bar{o})$ be the optimal solution of the Lagrangian dual problem for $\omega \in \Omega_{SP}$; then,

$$\theta(\omega) \geq \sum_{j \in V^+} \left(\phi \left(\bar{e_j}(\omega) + \bar{l_j}(\omega) \right) + \psi \bar{o}(\omega) + \lambda_j (t_i^s - \bar{q}_j^s) + \eta_j (t_j^e - \bar{q}_j^e) \right) + \sum_{i \in V} \sum_{j \in V} \beta_{ij} (x_{ij} - \bar{z}_{ij})$$

is a valid strengthened optimality cut for the MP.

5. Performance of TBDS

We show the performance of the Two-Step Benders Decomposition with Scenario Clustering (TBDS) method by solving the Time Window Assignment Traveling Salesperson Problem with Stochastic Travel Times (TWATSP-ST). We develop six variants of TBDS to assess and structurally benchmark the two main novelties of TBDS: the guided selection of scenarios in the master problem and the two-step decomposition of binary and linear first-stage variables. The six considered variants are:

1. **BD** uses *strengthened optimality cuts* and a standard decomposition over scenarios, leading to Benders dual decomposition, as introduced by Rahmaniani et al. (2020).
2. **TBD** extends BD by including the decomposition over the integer and linear first-stage variables. This variant tests the impact of our two-step first-stage decomposition compared with Benders dual decomposition.
3. **BDP** extends BD by including the first-stage constraints on the master problem of randomly chosen scenarios and artificial scenarios. This variant combines partial Benders decomposition (Crainic et al. 2021) with Benders dual decomposition.
4. **TBDP** extends BDP by including the two-step first-stage decomposition.
5. **BDS** extends BDP by including the guided selection of scenarios in the master program.
6. **TBDS** extends BDS by including the two-step first-stage decomposition. This is our TBDS method as presented in Section 3.

We evaluate the performance of the variants on a new set of benchmark instances introduced in Section 5.1. We carefully optimize the hyperparameters associated with the different variants, as detailed in Section 5.2. To measure the effectiveness of the proposed variants, we first compare the quality of the root node lower and upper bounds in Section 5.3. We continue by showcasing the full performance of the variants in Section 5.4, the convergence behavior of our method by conducting an overall branch-and-cut search and comparing the performance of the six variants for benchmark purposes. Finally, we provide insights into the optimal solution structure and the value of the stochastic solution. We derive managerial insights useful for practitioners in Section 5.5.

5.1. Benchmark Instances

We adapt benchmark instances from the literature to our problem setting, generating 126 new benchmark instances. Specifically, we derive 56 instances with clustered customer locations based on single vehicle routes from Solomon's VRPTW-RC instances proposed by Potvin and Bengio (1996) and 70 instances from Gendreau et al. (1998), which include not only the customer locations but also the associated service times. We refer to these as *rc_* and *n_w_* instances, respectively. The end of the depot's time window defines the shift length. All instances and associated solutions are included as supplementary material to the submission.

To balance the four penalty terms in the objective function and make them directly comparable concerning the routing cost for most problem instances, we used penalty weights of $\phi = 3$ and $\varphi = 1$ to penalize expected delay/earliness and time window width at customer $i \in V^+$, respectively. Comparably to Jabali et al. (2015), we set the penalty for expected shift overtime to $\psi = 4$.

We conduct a Sample Average Approximation analysis to determine the number of scenarios to correctly represent uncertainty, considering our routing solutions' stability as the number of customers in the instances increases. This results in 100 scenarios, which we use throughout all experiments.

We follow a similar approach for generating random travel times as Jabali et al. (2015) and Vareias, Repoussis, and Tarantilis (2019). For each scenario, we determine the travel time t_{ij} for arc $(i,j) \in A$ by adding a random disruption parameter δ_{ij} to the Euclidean distance d_{ij}. To model realistic travel times and disruptions, we assume a Gamma-distributed disruption parameter with shape k and scale θ_{ij} depending on the distance and a coefficient of variation $cov = 0.25$. We define $\eta = 0.35$ as a congestion level, representing the expected increase in travel time after a disruption occurs, i.e., $\mathbb{E}[\delta_{ij}] = \eta d_{ij}$. We assume $\delta_{ij} \sim G(k, \theta_{ij})$, resulting in

$$\mathbb{E}[\delta_{ij}] = k\theta_{ij} = \eta d_{ij},$$
$$Var(\delta_{ij}) = k\theta_{ij}^2.$$

Here, parameters k and θ_{ij} are set according to

$$k = \frac{1}{cov^2}, \quad \theta_{ij} = \eta d_{ij} cov^2$$

5.2. Parameter Settings and Implementation Details

The master problem (41) - (52) contains both a set of selected scenarios from Ω and some artificial scenarios created based on Ω. Section 3.2 explains in detail how to select scenarios and create artificial scenarios. The performance of our method depends on the number of scenarios added to the master program. By varying the fraction of scenarios added to the master program between 5%, 10% and 15%, we evaluate the performance of the methods BDP, TBDP, BDS, and TBDS under different scenario selection strategies. The results presented for the aforementioned methods are obtained by selecting the best combination of scenario levels, corresponding to the highest number of solved instances in each method.

We embed all methods in a branch-and-cut framework. We only add Benders cuts (20) - (23) at the root node and upon discovery of a possible incumbent solution. Subtour elimination constraints are included dynamically. All algorithms are coded in Python 3.9.7 in combination with Gurobi 9.5.1. The experiments are run on a virtual machine with 64 GB RAM under the Linux operating system, which was sufficient for all experiments. All algorithms have a time limit of three hours.

5.3. Performance of TBDS at the Root Node

Table 1 and 2 compare lower bounds (LB), upper bounds (UB), and root node gaps of the six TBDS variants (*BD, TBD, BDP, TBDP, BDS, TBDS*) for different instance sizes (10, 13, 15, 18, 20, 23, 25), listed in the first column. The number of instances solved with different customer numbers and the average lower and upper bound value for both benchmark sets is reported in Tables 1 and 2. The *Root Node Gap* is computed as $((UB - LB)/LB) \times 100)$ and gives the relative difference between lower and upper bound at the root node. The lower and upper bound information is obtained as the root node within the evaluation of the full branch-and-cut algorithm after the first branching decision, which partly depends on the procedures embedded within Gurobi.

Comparing the results in Tables 1 and 2, a few observations stand out. First, comparing TBD to BD, the upper bounds in TBD decrease substantially, with a reduction of 83.1% and 90.9% compared to BD for the *n_w_* and *rc_* instances, respectively.

Second, the concept of partial Benders decomposition helps to tighten the lower bound at the root node, i.e., comparing BDP and TBDP with BD and TBD. It alleviates the primal inefficiencies with redundant solutions in early iterations of the BD and TBD methods. The addition of any scenarios improves the lower and upper bound on average by 19.8% and 72.5% over the *n_w_* instances, respectively. For the *rc_* instances, the lower bound improves on average by 12.3% with the addition of any scenarios. Similarly, the upper bound decrease by 46.6% for the *rc_* instances.

We observe the benefit of choosing scenarios with clustering as it tightens the lower bound on average by 32.6% for *n_w_* instances and 5.6% for *rc_* instances comparing the BDS and TBDS methods to the BDP and TBDP methods. Comparing TBDP with TBDS, we see that in TBDS, on average, the lower bound increases by 22.5%, and the upper bound decreases by 35.5%.

As the number of customers increases in both instances, the benefits gained from scenario clustering and the two-step decomposition become more pronounced, although the root gap remains high. In the next section, we will analyze the impact of these findings on computational effort within the full branch-and-cut procedure.

Table 1 Average Root Node Lower and Upper Bounds of n_w_ Instances for the TBDS Variants

# Customers	Root Node	BD	TBD	BDP	TBDP	BDS	TBDS
	Lower Bound	195.30	200.95	228.22	245.72	234.15	240.68
10	Upper Bound	5586.13	252.78	343.83	278.73	288.98	259.27
	Root Node Gap (%)	96.50	20.50	33.62	11.84	18.97	7.17
	Lower Bound	225.89	237.74	245.15	243.45	296.73	311.16
13	Upper Bound	9914.30	352.71	2451.89	402.81	381.90	345.47
	Root Node Gap (%)	97.72	32.60	90.00	39.56	22.30	9.93
	Lower Bound	231.87	264.84	291.49	272.59	331.45	332.53
15	Upper Bound	13446.87	538.15	5258.36	1172.17	461.39	444.73
	Root Node Gap (%)	98.28	50.79	94.46	76.74	28.16	25.23
	Lower Bound	254.25	256.70	304.22	313.66	387.24	365.03
18	Upper Bound	18862.60	1932.04	1397.69	1770.36	1439.92	1019.40
	Root Node Gap (%)	98.65	86.71	78.23	82.28	73.11	64.19
	Lower Bound	274.07	316.01	415.77	321.01	383.73	383.20
20	Upper Bound	23932.42	3041.61	2614.86	2017.36	1856.67	1440.71
	Root Node Gap (%)	98.85	89.61	84.10	84.09	79.33	73.40
	Lower Bound	291.82	353.40	409.85	346.43	689.14	694.77
23	Upper Bound	41065.30	7002.24	3498.46	6933.25	3793.50	1949.43
	Root Node Gap (%)	99.29	94.95	88.28	95.00	81.83	64.36
	Lower Bound	313.63	391.14	529.54	392.09	673.76	721.30
25	Upper Bound	43749.30	13324.17	20787.50	1452.33	5733.92	2564.44
	Root Node Gap (%)	99.28	97.06	97.45	73.00	88.25	71.87

Table 2 Average Root Node Lower and Upper Bounds of rc_ Instances for All TBDS Variants

# Customers	Root Node	BD	TBD	BDP	TBDP	BDS	TBDS
	Lower Bound	363.35	363.35	369.38	391.62	399.29	401.83
10	Upper Bound	5988.77	409.21	6625.27	461.50	450.11	458.53
	Root Node Gap (%)	93.93	11.21	94.42	15.14	11.01	12.37
	Lower Bound	390.74	362.55	450.48	462.54	487.65	482.45
13	Upper Bound	8818.20	420.75	11157.97	562.48	617.31	576.07
	Root Node Gap (%)	95.57	13.83	95.96	17.77	21.00	16.25
	Lower Bound	423.23	438.95	476.32	513.39	528.96	523.52
15	Upper Bound	12160.68	509.61	13447.64	605.66	654.87	650.58
	Root Node Gap (%)	96.52	13.87	96.46	15.23	19.23	19.53
	Lower Bound	506.59	528.65	536.37	587.94	600.11	601.10
18	Upper Bound	19988.22	681.00	21811.96	903.21	778.95	715.95
	Root Node Gap (%)	97.47	22.37	97.54	34.91	22.96	16.04
	Lower Bound	538.36	582.28	651.14	644.84	674.16	672.40
20	Upper Bound	22563.14	1832.02	3747.24	1081.68	1078.71	1347.28
	Root Node Gap (%)	97.61	68.22	82.62	40.39	37.50	50.09
	Lower Bound	595.36	626.84	628.14	735.01	724.77	720.46
23	Upper Bound	32158.67	2978.55	1765.22	2174.53	2220.65	2756.16
	Root Node Gap (%)	98.15	78.95	64.42	66.20	67.36	73.86
	Lower Bound	628.92	679.91	710.37	734.75	764.11	755.07
25	Upper Bound	37700.76	5879.10	10538.78	6334.71	31067.64	6091.77
	Root Node Gap (%)	98.33	88.44	93.26	88.40	97.54	87.61

5.4. Overall Performance of TBDS

We continue by analyzing the overall performance of the six TBDS variants on our benchmark instances. Our proposed solution approach, TBDS, solves up to 25 customers in both sets of benchmark instances. Looking to the total number of solved instances within the time limit displayed in Figure 1, it is evident that the variants with two-step decomposition outperform the variants without two-step decomposition. Specifically, TBDS solved 90.0% and 85.7% of instances in both benchmark sets to optimality, while BDS only solved 28.6% and 26.8% of instances. In contrast, BD failed to solve any instance, and TBD only solved up to 5.7% and 3.6% of all instances. The BDP and TBDP variants were able to solve between 21.4% and 48.3% of all benchmark instances. The significant increase in performance of TBDS can be attributed to the use of smart clustering and the two-step decomposition over continuous and binary first-stage variables.

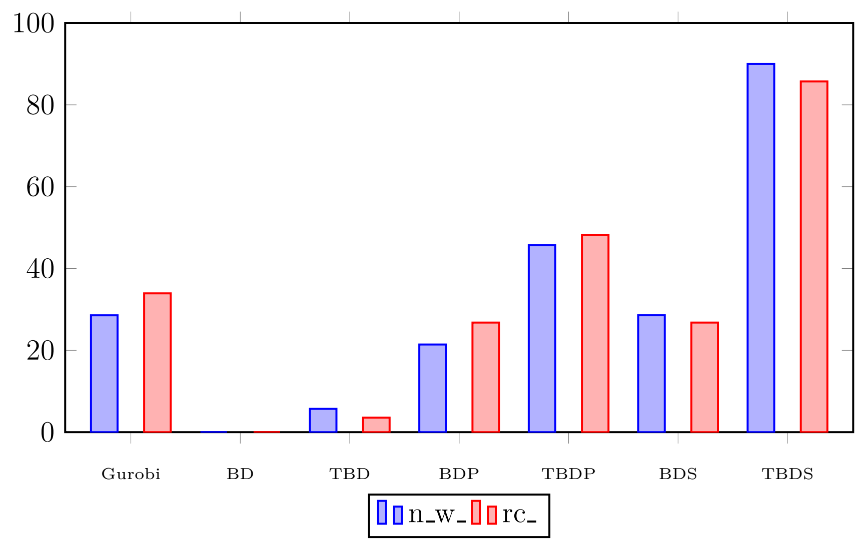


Figure 1 Percentage of Instances Solved Within 3 hours

Tables 3 and 4 provide detailed results for each benchmark set. Noting that the BD variant fails to solve any instance within the 3 hours time limit, the computational time of solved instances in the variants without the two-step decomposition ranges between 127.8 minutes and 149.5 minutes. We see a clear decrease in computational time ranging between 73.2 and 99.7 minutes in the variants with the two-step decomposition. Comparing BD, BDP and BDS, the average optimality gap of non-solved instances decreases significantly. Among variants without two-step decomposition, the average optimality gap decreases from 58.5% to 36.6% to 15.8%. Similarly, with the variants with two-step decomposition, we see that the average optimality gap decreases from 30.5% to 5.8%. The results clearly show that our TBDS outperforms the state-of-the-art, as it solves more instances to

Table 3 Performance of the Six TBDS variants on n_w_ instances

# Customers		BD	TBD	BDP	TBDP	BDS	TBDS
	Solved (#/10)	0	4	5	10	5	10
10	Time to Optimality	-	30.05	34.86	20.97	108.80	3.27
	Opt. Gap Unsolved	26.44	0.37	1.64	-	1.12	-
	Solved (#/10)	0	0	7	10	7	10
13	Time to Optimality	-	-	85.93	62.65	109.06	29.69
	Opt. Gap Unsolved	27.76	1.27	20.43	0.00	3.12	-
	Solved (#/10)	0	0	2	7	2	10
15	Time to Optimality	-	-	152.47	93.20	170.13	36.49
	Opt. Gap Unsolved	45.71	2.61	20.18	0.081	7.54	0.00
	Solved (#/10)	0	0	1	3	4	10
18	Time to Optimality	-	-	164.17	159.46	116.85	70.85
	Opt. Gap Unsolved	77.30	14.64	32.83	16.88	15.13	0.00
	Solved (#/10)	0	0	0	2	2	10
20	Time to Optimality	-	-	-	131.40	127.47	99.22
	Opt. Gap Unsolved	81.04	45.40	44.54	38.17	35.33	0.00
	Solved (#/10)	0	0	0	0	0	8
23	Time to Optimality	-	-	-	-	-	135.25
	Opt. Gap Unsolved	92.45	58.56	60.09	24.72	26.13	3.39
	Solved (#/10)	0	0	0	0	0	5
25	Time to Optimality	-	-	-	-	-	97.71
	Opt. Gap Unsolved	97.50	75.38	73.59	70.96	37.74	11.67

Table 4 Performance of the Six TBDS variants on rc_ instances

# Customers		BD	TBD	BDP	TBDP	BDS	TBDS
	Solved (#/8)	0	2	6	8	6	8
10	Time to Optimality	-	149.79	135.92	31.25	165.19	6.33
	Opt. Gap Unsolved	13.32	7.19	13.15	-	7.36	-
	Solved (#/8)	0	0	6	8	6	8
13	Time to Optimality	-	-	104.88	82.81	179.21	25.30
	Opt. Gap Unsolved	27.25	8.20	16.05	-	7.23	-
	Solved (#/8)	0	0	2	8	3	8
15	Time to Optimality	-	-	167.48	128.15	173.37	38.44
	Opt. Gap Unsolved	20.03	4.62	27.57	-	7.54	-
	Solved (#/8)	0	0	1	1	0	8
18	Time to Optimality	-	-	176.52	155.62	-	77.99
	Opt. Gap Unsolved	56.42	42.28	20.17	3.12	8.10	-
	Solved (#/8)	0	0	0	2	0	8
20	Time to Optimality	-	-	-	131.27	-	93.37
	Opt. Gap Unsolved	65.25	21.67	47.46	12.89	7.82	-
	Solved (#/8)	0	0	0	0	0	5
23	Time to Optimality	-	-	-	-	-	139.62
	Opt. Gap Unsolved	92.20	64.27	66.12	11.06	14.46	2.27
	Solved (#/8)	0	0	0	0	0	3
25	Time to Optimality	-	-	-	-	-	170.71
	Opt. Gap Unsolved	95.61	80.87	67.89	12.12	43.19	6.02

optimality and obtains smaller optimality gaps for the instances unable to solve optimally. Essential is the combination of the two main ideas of TBDS, i.e., the two-step decomposition and the guided selection of representative scenarios.

5.5. Managerial Insights

This section assesses the value of integrating time window assignment and routing decisions and provides insights into the structure of the optimal stochastic solutions. To do so, we fix different routing decisions and time window assignments, as well as both first-stage variables, and analyze the resulting solution costs. Further, we conduct analysis on the value of stochastic solution. Our insights quantify the benefits of optimizing routing and time window assignments simultaneously (Insight 1), assigning customer-specific time windows (Insight 2) and considering stochasticity in optimizing routing and time window assignment (Insights 3 and 4).

Insight 1. *Simultaneously optimizing time windows and routing decreases the expected time window exceedance by 73.0% and reduces the width of assigned time windows by 6.9% while routing costs only increase slightly (5.2%).*

Tables 5 and 6 summarize the routing decisions and associated costs for various instance sizes (10, 13, 15, 18, 20), considering three solution types. The Mean Value Problem (MVP) solution is obtained by optimizing both routing and time window assignment assuming travel times follow their expectation. The routing solution obtained is then fixed and used as input in our model subject to all scenarios. The Traveling Salesperson Solution (TSP) simply fixes the routing according to the shortest tour and uses this as input in our model subject to all scenarios. The Stochastic Solution (SS) is the optimal solution to the TWATSP-ST as defined before. The column *Routing Cost* gives the resulting routing cost, the column *TWA Cost* presents the time window assignment cost, and column *Recourse Cost* the second-stage costs.

Our analysis reveals that even small variations in routing decisions yield significant reductions in both time window width and recourse costs. Moreover, the TSP solution performs poorly in the second stage, particularly as the number of customers increases. That suggests that wider time window assignments do not necessarily result in better performance in the second stage. With more customers, the stochastic solution's cost advantage over MVP and TSP solutions becomes more pronounced.

Insight 2. *The flexibility to vary the time window width among the customers in the tour decreases the total time window width by 18.7% and decreases total cost by 7.8% on average. 67.2% of customers receive narrower time windows compared to assigning only fixed-width ones.*

Table 5 Analysis of different routing decisions in n_w_ instances

# Customers	Solution Type	Routing Cost	TWA Cost	Recourse Cost	Total Cost
10	MVP	151.10	75.17	39.00	265.28
	SS	153.37	58.98	33.94	246.28
	TSP	151.10	74.38	38.40	263.88
13	MVP	171.78	100.93	57.42	330.12
	SS	186.68	78.10	49.94	314.72
	TSP	171.47	99.82	90.03	361.32
15	MVP	177.44	114.92	62.82	355.18
	SS	183.66	92.69	59.14	335.49
	TSP	175.73	118.85	299.31	593.88
18	MVP	199.14	141.72	76.22	417.08
	SS	212.95	114.36	70.80	398.11
	TSP	191.29	132.28	753.99	1077.56
20	MVP	197.03	152.48	82.25	431.76
	SS	205.33	130.25	83.91	419.49
	TSP	194.27	149.18	223.85	567.30

Table 6 Analysis of different routing decisions in rc_ instances

# Customers	Solution Type	Routing Cost	TWA Cost	Recourse Cost	Total Cost
10	MVP	167.35	213.34	28.53	409.22
	SS	170.23	209.96	27.72	407.91
	TSP	167.35	216.58	30.53	414.56
13	MVP	181.80	284.79	51.26	517.85
	SS	192.99	272.80	37.63	503.43
	TSP	181.80	283.60	48.79	514.18
15	MVP	191.34	314.89	63.47	569.70
	SS	199.67	302.81	45.52	548.01
	TSP	191.34	310.31	56.45	558.10
18	MVP	208.51	366.57	76.53	651.62
	SS	216.10	354.20	56.00	626.29
	TSP	208.11	366.04	110.54	684.69
20	MVP	220.80	411.07	79.33	711.20
	SS	225.48	397.64	70.38	693.50
	TSP	218.28	410.06	332.28	960.62

Table 7 shows that for small instances with 10 to 13 customers, we compare fixed time window widths (called FTWAS) with the stochastic solution. The FTWAS policy fixes the time window width to the mean of the time window assignments in the stochastic solution. The associated costs are reported in Table 7. Customers receive at most 11.7% wider time windows and at least 7.3% narrower time windows than the fixed time windows. The results highlight the importance of assigning time windows individually rather than providing the same interval for every customer.

Table 7 Analysis of fixed time window assignment

	Solution Type	Routing Cost	TWA Cost	Recourse Cost	Total Cost
n_w_	SS	155.55	66.08	36.52	258.16
	FTWAS	156.13	74.95	33.60	264.68
rc_	SS	184.25	215.61	30.46	430.33
	FTWAS	184.36	271.52	26.83	482.72

Insight 3. *Incorporating travel time uncertainty into the optimization of routing and time window assignments decreases costs by 5.7%. This benefit stems from improving on-time delivery by 51.8% at the expense of a 3.3% increase in routing cost and 14.6% wider time windows.*

Tables 8 and 9 show the routing, time window assignment, second-stage and total cost of expected mean value problem (EMVP) solution and stochastic solution (SS) for different instances. The EMVP is obtained by considering only a single scenario representing the expected travel times and afterwards evaluating this solution on each of the scenarios. The difference between SS and EMVP gives the value of the stochastic solution (VSS). Our results indicate that incorporating different travel time scenarios into decision-making – through the use of SS – results in an average VSS of 8.1% and 4.3% for *n_w_* and *rc_* instances, respectively. Additionally, our analysis reveals that a slight increase in routing cost and wider time window assignments leads to better performance in terms of on-time delivery in the second stage.

Insight 4. *Optimal routes often exhibit an alternating pattern in time window width and variance of incoming arcs. As such, the optimal solution hedges for time window violations with buffer times distributed throughout the route.*

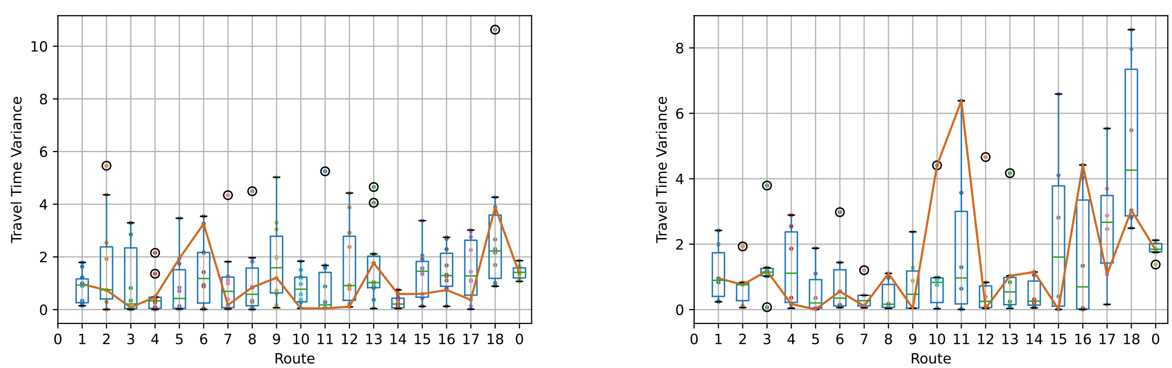
We further analyze the nature of routing and time window assignment solutions generated by TBDS. We observe that optimal routes exhibit a zigzag pattern with high variations and large time

Table 8 Comparison of Expected Mean Value Problem (EMVP) solution with Stochastic solution (SS) in n_w_ instances

# Customers	Math. Model	Routing Cost	TWA Cost	Recourse Cost	Total Cost
10	EMVP	151.10	50.00	71.05	272.15
	SS	153.37	58.98	33.94	246.28
13	EMVP	171.78	65.00	102.45	339.22
	SS	186.68	78.10	49.94	314.72
15	EMVP	177.44	75.00	113.08	365.52
	SS	183.66	92.69	59.14	335.49
18	EMVP	199.14	90.00	141.23	430.37
	SS	212.95	114.36	70.80	398.11
20	EMVP	197.03	100.00	148.29	445.33
	SS	205.33	130.25	83.91	419.49

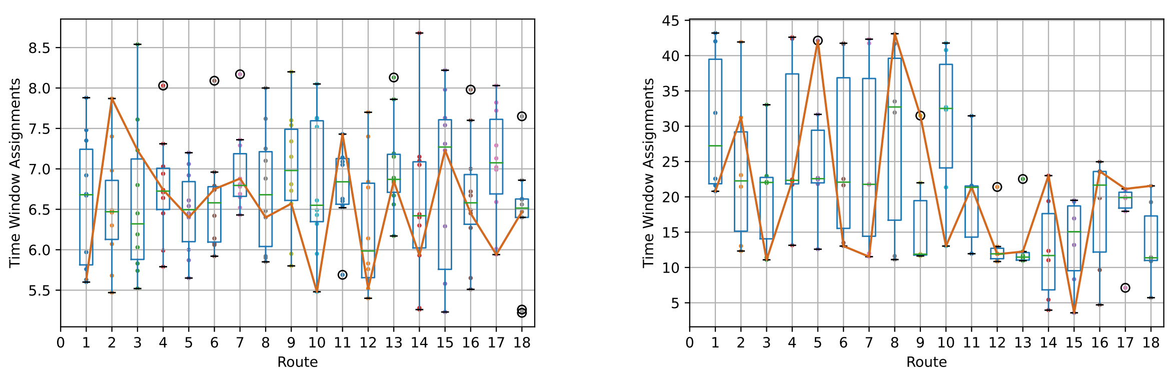
Table 9 Comparison of Expected Mean Value Problem (EMVP) solution with Stochastic solution (SS) in rc_ instances

# Customers	Math. Model	Routing Cost	TWA Cost	Recourse Cost	Total Cost
10	EMVP	176.01	194.92	50.80	421.74
	SS	170.23	209.96	27.72	407.91
13	EMVP	191.74	224.50	91.44	507.68
	SS	192.99	272.80	37.63	503.43
15	EMVP	191.34	275.71	112.81	579.87
	SS	199.67	302.81	45.52	548.01
18	EMVP	208.51	320.86	133.78	663.15
	SS	216.10	354.20	56.00	626.29
20	EMVP	220.80	359.17	144.65	724.62
	SS	225.48	397.64	70.38	693.50



(a) Travel time variance in *n_w_* instances **(b)** Travel time variance in *rc_* instances

Figure 2 Travel time variance of optimal routing solutions



(a) Time window assignments in *n_w_* instances **(b)** Time window assignments in *rc_* instances

Figure 3 Time window assignment solutions in *rc_* instances windows alternating with smaller variations and smaller time windows. Figure 2 shows the variance of the travel time of arcs within the optimal routing sequence for *n_w_* and *rc_* instances with 18 customers. Similarly, Figure 3 provides the optimal width of the time windows. An orange line highlights an example instance to illustrate the pattern more clearly. As expected, customers reached using arcs with high variance receive wider time windows. As such, the model immediately hedges for delays. The optimal solution combines risky and less risky choices. Like this, delays do not propagate too severely through the route. Observing – or constructing and utilizing – these alternating patterns is impossible with current heuristics, which ignore delay propagation. The time windows are more evenly distributed in the *nw_* instances than in the *rc_* (see Figure 3), indicating a balance between risky and non-risky customer choices. This pattern also explains Insights 1 and 2.

6. Conclusions

We present a new method called Two-Step Benders Decomposition with Scenario Clustering (TBDS) for solving two-stage stochastic mixed-integer programs. Our method combines and generalizes the recent advancements in Benders decomposition and scenario clustering techniques. Our TBDS method introduces a novel two-step decomposition strategy for the binary and continuous first-stage variables, resulting in improved continuous first-stage solutions while generating optimality cuts. This two-step decomposition approach leads to high-quality initial first-stage solutions, effectively reducing unnecessary iterations that typically occur in current state-of-the-art Benders decomposition approaches. Consequently, it enhances computational efficiency by facilitating faster convergence.

The second key contribution of TBDS is incorporating clustered scenarios into the master program, which is, to the best of the authors' knowledge, the first time that such scenario clustering techniques and state-of-the-art Benders decomposition approaches are combined. By clustering the scenarios, we improve the linear programming (LP) relaxation of the master problem, obtaining superior lower bounds in the early iterations. By combining these two essential elements (i.e., the two-step decomposition and the scenario clustering), our method achieves consistently tighter bounds at the root node and produces higher quality incumbent solutions compared to state-of-the-art approaches in the extant literature, including Benders dual decomposition and partial Benders decomposition. Specifically, these methods can be considered special cases of TBDS.

We use TBDS to solve the Time Window Assignment Problem with Stochastic Travel Times (TWATSP-ST), a challenging combinatorial problem formulated as a two-stage stochastic mixed-integer program with continuous recourse for which no efficient exact solution methods exist yet. Extensive experimental results demonstrate the effectiveness of TBDS. Our method solves more instances to optimality, and significantly better lower and upper bounds are obtained for the instances not solved to optimality. In particular, TBDS achieves optimality for 87.9% of the instances in our benchmark set, surpassing other benchmark algorithms that can solve at most 47% of the instances. This showcases the superior performance and efficiency of our TBDS method in handling TWATSP-ST. Furthermore, our study reveals that the simultaneous optimization of time windows and routing leads to a noteworthy 12.8% improvement in total costs while incurring only a minor increase in routing costs. Allowing different time window lengths enables hedging against high variances encountered throughout the route. As a result, our method produces shorter routes with fewer time window violations, contributing to the overall cost reduction.

In future studies, the alternating pattern of high and low variance arcs traveled, as observed in the structure of the optimal solution, can serve as a foundation for developing efficient heuristics tailored specifically for the TWATSP-ST. Additionally, evaluating our algorithm on extended versions of the vehicle routing problem, such as the capacitated vehicle routing problem and the multi-depot vehicle routing problem, can offer valuable insights into the versatility and applicability of our TBDS method in diverse real-world scenarios.

From a methodological perspective, we envision that the problem-specific selection of representative scenarios in combination with Benders decomposition approaches can be the start of several new research lines. For instance, using supervised learning to predict which scenarios to label as representative based on instance-specific information such as vehicle information and customer locations seems promising. Especially for applications with limited information, it would be valuable to research how well such predictions translate to slightly different settings. Alternatively, methods other than those in TBDS can be developed and tested for generating representative scenarios.

Acknowledgments

Albert H. Schrotenboer has received support from the Dutch Science Foundation (NWO) through grant VI.Veni.211E.043

References

Benders J, 1962 *Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik* 4(1):238–252.

Crainic TG, Hewitt M, Maggioni F, Rei W, 2021 *Partial Benders decomposition: General methodology and application to stochastic network design. Transportation Science* 55(2):414–435.

Cui Z, Long DZ, Qi J, Zhang L, 2023 *The inventory routing problem under uncertainty. Operations Research* 71(1):378–395.

Dantzig GB, 1955 *Linear programming under uncertainty. Management Science* 1(3-4):197–206.

Ehmke JF, Campbell AM, Urban TL, 2015 *Ensuring service levels in routing problems with time windows and stochastic travel times. European Journal of Operational Research* 240(2):539–550.

Gendreau M, Hertz A, Laporte G, Stan M, 1998 *A generalized insertion heuristic for the traveling salesman problem with time windows. Operations Research* 46(3):330–335.

Henrion R, Küchler C, Römisch W, 2009 *Scenario reduction in stochastic programming with respect to discrepancy distances. Computational Optimization and Applications* 43(1):67–93.

Henrion R, Römisch W, 2018 *Problem-based optimal scenario generation and reduction in stochastic programming. Mathematical Programming* 1–23.

Jabali O, Leus R, Van Woensel T, De Kok T, 2015 *Self-imposed time windows in vehicle routing problems. OR Spectrum* 37(2):331–352.

Keutchayan J, Munger D, Gendreau M, 2020 *On the scenario-tree optimal-value error for stochastic programming problems. Mathematics of Operations Research* 45(4):1572–1595.

Keutchayan J, Ortmann J, Rei W, 2023 *Problem-driven scenario clustering in stochastic optimization. Computational Management Science* 20(1):13.

Kleywegt AJ, Shapiro A, Homem-de Mello T, 2002 *The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization* 12(2):479–502.

Paradiso R, Roberti R, Laganá D, Dullaert W, 2020 *An exact solution framework for multitrip vehicle-routing problems with time windows. Operations Research* 68(1):180–198.

Pflug GC, Pichler A, 2015 *Dynamic generation of scenario trees. Computational Optimization and Applications* 62(3):641–668.

Potvin JY, Bengio S, 1996 *The vehicle routing problem with time windows part ii: genetic search. INFORMS Journal on Computing* 8(2):165–172.

Rahmaniani R, Ahmed S, Crainic TG, Gendreau M, Rei W, 2020 *The Benders dual decomposition method. Operations Research* 68(3):878–895.

Rahmaniani R, Crainic TG, Gendreau M, Rei W, 2017 *The benders decomposition algorithm: A literature review. European Journal of Operational Research* 259(3):801–817.

Schiffer M, Schneider M, Walther G, Laporte G, 2019 *Vehicle routing and location routing with intermediate stops: A review. Transportation Science* 53(2):319–343.

Slyke RMV, Wets R, 1969 *L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics* 17(4):638–663.

Spliet R, Dabia S, Van Woensel T, 2018 *The time window assignment vehicle routing problem with time-dependent travel times. Transportation Science* 52(2):261–276.

Van Slyke RM, Wets R, 1969 *L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics* 17(4):638–663.

Vareias AD, Repoussis PP, Tarantilis CD, 2019 *Assessing customer service reliability in route planning with self-imposed time windows and stochastic travel times. Transportation Science* 53(1):256–281.

Wölck M, Meisel S, 2022 *Branch-and-price approaches for real-time vehicle routing with picking, loading, and soft time windows. INFORMS Journal on Computing* 34(4):2192–2211.

Appendix A: Mixed-Integer Program for Minimizing the Clustering Error

$$\min \quad \frac{1}{|\Omega|}\sum_{i\in\Omega} t_i \tag{58}$$

$$\text{s.t.} \quad t_j \geq \sum_{i\in\Omega}\sigma_{ij}V_{ji} - \sum_{i\in\Omega}\sigma_{ij}V_{jj}, \qquad \forall\, j\in\Omega, \tag{59}$$

$$t_j \geq \sum_{i\in\Omega}\sigma_{ij}V_{jj} - \sum_{i\in\Omega}\sigma_{ij}V_{ji}, \qquad \forall\, j\in\Omega, \tag{60}$$

$$\sigma_{ij} \leq u_j, \quad \sigma_{jj} = u_j \qquad \forall (i,j)\in\Omega\times\Omega, \tag{61}$$

$$\sum_{j\in\Omega}\sigma_{ij} = 1, \quad \sum_{j\in\Omega} u_j = K \qquad \forall\, i\in\Omega, \tag{62}$$

$$\sigma_{ij}\in\{0,1\}, u_j\in\{0,1\}, t_i\in\mathbb{R}_+ \qquad \forall (i,j)\in\Omega\times\Omega. \tag{63}$$

We define binary variable u_j to determine if a scenario $j\in\Omega$ is picked as a cluster representative and another binary variable σ_{ij} to identify scenario $i\in\Omega$ if it is in the cluster with representative scenario $j\in\Omega$. Constraint (59)-(60) linearize the equation (25). Constraint (61) and (62) ensure that we construct K non empty clusters that contains a representative scenario.

Appendix B: SP1 for the TWATSP-ST

For a feasible solution x^*, SP1 for TWATSP-ST is formulated as follows:

$$SP1(x^*) = \min \quad \sum_{j\in V^+}\varphi\left(t_i^e - t_i^s\right) + \sum_{\omega\in\Omega_{SP}} p_\omega \left(\sum_{j\in V^+}\phi\left(e_j(\omega) + l_j(\omega)\right) + \psi o(\omega)\right)$$

$$\text{s.t.} \quad w_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) + s_j - (1-x_{ij}^*)M \qquad \forall i\in V, j\in V, \omega\in\Omega_{SP}, \quad [\lambda^1]$$

$$e_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) - t_i^s - (1-x_{ij}^*)M \qquad \forall i\in V, j\in V, \omega\in\Omega_{SP}, \quad [\lambda^2]$$

$$l_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) + s_j - t_i^e - (1-x_{ij}^*)M \qquad \forall i\in V, j\in V, \omega\in\Omega_{SP}, \quad [\lambda^3]$$

$$o(\omega) \geq w_i(\omega) + t_{i0}(\omega) - T \qquad \forall i\in V^+, \omega\in\Omega_{SP}, \quad [\mu]$$

$$t_i^e - t_i^s \geq s_i \qquad \forall i\in V^+, \quad [\nu]$$

$$w_0(\omega) = t_0 \qquad \forall \omega\in\Omega_{SP}, \quad [\pi]$$

$$t_i^s, t_i^e \in\mathbb{R}_+ \qquad \forall i\in V^+,$$

$$w_i(\omega), e_i(\omega), l_i(\omega)\in\mathbb{R}_+ \qquad \forall i\in V, \omega\in\Omega_{SP},$$

$$o(\omega)\in\mathbb{R}_+ \qquad \forall \omega\in\Omega_{SP}.$$

Let the letters next to each constraint in the SP1 be a dual variable associated with the corresponding constraint.

Appendix C: SP2 for the TWATSP-ST

We define the second set of subproblems $SP2(x^*, t^{s*}, t^{e*}, \omega)$ for a given feasible solution (x^*, t^{s*}, t^{e*}) and $\omega \in \Omega_{SP}$ as follows

$$
\begin{aligned}
\min \quad & \sum_{j \in V^+} \phi\left(e_j(\omega) + l_j(\omega)\right) + \psi o(\omega) \\
\text{s.t.} \quad & \sum_{i \in V \setminus \{j\}} z_{ij} = \sum_{i \in V \setminus \{j\}} z_{ji} = 1 && \forall j \in V, \\
& z_{ii} = 0 && \forall i \in V, \\
& \sum_{i \in S} \sum_{j \notin S} z_{ij} \geq 1 && S \subseteq V, 1 \leq |S| \leq |V^+|, \\
& w_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) + s_j - (1 - z_{ij})M && \forall i \in V, j \in V, \\
& e_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) - q_i^s - (1 - z_{ij})M && \forall i \in V, j \in V, \\
& l_j(\omega) \geq w_i(\omega) + t_{ij}(\omega) + s_j - q_i^e - (1 - z_{ij})M && \forall i \in V, j \in V, \\
& o(\omega) \geq w_i(\omega) + t_{i0}(\omega) - T && \forall i \in V^+, \\
& q_i^e - q_i^s \geq s_i && \forall i \in V^+, \\
& w_0(\omega) = t_0, \\
& z_{ij} = x_{ij}^* && \forall i \in V, j \in V, \quad [\beta] \\
& q_i^s = t_i^{s*} && \forall i \in V^+, \quad [\lambda] \\
& q_i^e = t_i^{e*} && \forall i \in V^+, \quad [\eta] \\
& z_{ij} \in \mathbb{R}_+ && \forall i \in V, j \in V, \\
& q_i^s, q_i^e \in \mathbb{R}_+ && \forall i \in V^+, \\
& w_i(\omega), e_i(\omega), l_i(\omega) \in \mathbb{R}_+ && \forall i \in V, \\
& o(\omega) \in \mathbb{R}_+.
\end{aligned}
$$

Appendix D: Feasibility Problem For TWATSP-ST

For infeasible x^* solution for SP1, we solve the following feasibility problem for $\omega \in \Omega_{SP}$.

$$
\begin{aligned}
\min \quad & \mathbb{1}^T(\epsilon^1 + \epsilon^2 + \epsilon^3) \\
\text{s.t.} \quad & \sum_{i \in V \setminus \{j\}} z_{ij} = \sum_{i \in V \setminus \{j\}} z_{ji} = 1 && \forall j \in V, \\
& z_{ii} = 0 && \forall i \in V, \\
& \sum_{i \in S} \sum_{j \notin S} z_{ij} \geq 1 && S \subseteq V, 1 \leq |S| \leq |V^+|, \\
& w_j(\omega) + \epsilon^1 \geq w_i(\omega) + t_{ij}(\omega) + s_j - (1 - z_{ij})M && \forall i \in V, j \in V, \omega \in \Omega_{SP}, \\
& e_j(\omega) + \epsilon^2 \geq w_i(\omega) + t_{ij}(\omega) - q_i^s - (1 - z_{ij})M && \forall i \in V, j \in V, \omega \in \Omega_{SP},
\end{aligned}
$$

$$l_j(\omega) + \epsilon^3 \geq w_i(\omega) + t_{ij}(\omega) + s_j - q_i^e - (1 - z_{ij})M \qquad \forall i \in V, j \in V, \omega \in \Omega_{SP},$$

$$o(\omega) \geq w_i(\omega) + t_{i0}(\omega) - T \qquad \forall i \in V^+, \omega \in \Omega_{SP},$$

$$q_i^e - q_i^s \geq s_i \qquad \forall i \in V^+,$$

$$w_0(\omega) = t_0 \qquad \forall \omega \in \Omega_{SP},$$

$$z_{ij} = x_{ij}^* \qquad \forall i \in V, j \in V, \quad [\lambda]$$

$$q_i^s = t_i^{s*} \qquad \forall i \in V^+,$$

$$q_i^e = t_i^{e*} \qquad \forall i \in V^+,$$

$$z_{ij} \in \mathbb{R}_+ \qquad \forall i \in V, j \in V,$$

$$q_i^s, q_i^e \in \mathbb{R}_+ \qquad \forall i \in V^+,$$

$$w_i(\omega), e_i(\omega), l_i(\omega) \in \mathbb{R}_+ \qquad \forall i \in V, \omega \in \Omega_{SP},$$

$$o(\omega) \in \mathbb{R}_+.$$