Benders Decomposition and Column-and-Row Generation for Solving Large-Scale Linear Programs with Column-Dependent-Rows

İbrahim Muter
School of Management, University of Bath, BA2 7AY Bath, UK
i.muter@bath.ac.uk

Ş. İlker Birbil, Kerem Bülbül
Sabancı University, Industrial Engineering, Orhanlı-Tuzla, 34956 Istanbul, Turkey
sibirbil@sabanciuniv.edu, bulbul@sabanciuniv.edu

Abstract: In a recent work, Muter et al. (2013a) identified and characterized a general class of linear programming (LP) problems - known as problems with column-dependent-rows (CDR-problems). These LPs feature two sets of constraints with mutually exclusive groups of variables in addition to a set of structural linking constraints, in which variables from both groups appear together. In a typical CDR-problem, the number of linking constraints grows very quickly with the number of variables, which motivates generating both columns and their associated linking constraints simultaneously on-the-fly. In this paper, we expose the decomposable structure of CDR-problems via Benders decomposition. However, this approach brings on its own theoretical challenges. One group of variables is generated in the Benders master problem, while the generation of the linking constraints is relegated to the Benders subproblem along with the second group of variables. A fallout of this separation is that only a partial description of the dual of the Benders subproblem is available over the course of the algorithm. We demonstrate how the pricing subproblem for the column generation applied to the Benders master problem does also update the dual polyhedron and the existing Benders cuts in the master problem to ensure convergence. Ultimately, a novel integration of Benders cut generation and the simultaneous generation of columns and constraints yields a brand-new algorithm for solving large-scale CDR-problems. We illustrate the application of the proposed method on a time-constrained routing problem. Our numerical experiments confirm the outstanding performance of the new decomposition method.

Keywords: large-scale optimization; linear programming; column-dependent-rows; column generation; column-and-row generation; Benders decomposition; pricing subproblem; quadratic set covering; time-constrained routing problem.

---

1. Introduction. We have recently developed a simultaneous column-and-row generation methodology (Muter et al., 2013a) for a general class of linear programming problems. The problems that belong to this class have an interesting structure. In their most general form, they feature two sets of constraints with mutually exclusive groups of variables in addition to a set of *linking* constraints, in which variables from both groups appear together. The list of applications which fit into this framework and satisfy the assumptions of our analysis include multi-stage cutting stock (Zak, 2002), P-median facility location (Avella et al., 2007), multi-commodity capacitated network design (Katayama et al., 2009; Frangioni and Gendron, 2009), two-stage batch scheduling (Wang and Tang, 2010), robust crew pairing (Muter et al., 2013b), and time-constrained routing (Avella et al., 2006; Muter et al., 2012). These problems are frequently formulated with too many variables to be included explicitly in the model at the outset and are therefore typically attacked by column generation techniques (Dantzig and Wolfe, 1960; Lübbecke and Desrosiers, 2005). The additional challenge here is that the number of linking constraints is either too large which precludes us from incorporating these constraints directly in the formulation, or an explicit description of the full set of linking constraints is only available in the presence of the entire set of variables. Therefore, whenever these problems are solved by column generation, the introduction of new

columns leads to the generation of new linking constraints. That is, the sequence of LPs solved during column generation does not only grow column-wise but also row-wise through the addition of new linking constraints. Consequently, we refer to this class of formulations as problems with *column-dependent-rows*, or more concisely as CDR-problems. A key point here is that these new linking constraints are structural constraints required for the validity of the formulation. This is a defining property which clearly distinguishes our work from the branch-and-cut-and-price setting as we elaborate upon in the next paragraph. The primary challenge in solving CDR-problems via column generation is to price out the columns absent from the formulation correctly as the dual variables associated with the missing linking constraints are unknown. To overcome this difficulty, we have developed a *thinking-ahead approach*, which computes the correct values of the dual variables of the missing linking constraints as if they were present in the formulation (Muter et al., 2013a). Recently, simultaneous generation of columns and rows has also been studied in other settings by Frangioni and Gendron (2013) and Sadykov and Vanderbeck (2013).

A large number of problems solved by column generation originally involve integrality constraints for some or all of the variables. To solve such problems to optimality, column generation is generally integrated into the branch-and-bound method giving rise to the branch-and-price method (Barnhart et al., 1998). Generating strengthened bounds at the nodes of the branch-and-price tree can be achieved through valid inequalities obtained by solving a separation subproblem. This use of column and row generation together in a branch-and-bound setting is known as branch-and-cut-and-price; see the works of Desaulniers et al. (2011) and Desrosiers and Lübbecke (2011). In the branch-and-cut-and-price framework, rows and columns are generated sequentially and independently from each other by solving the separation and pricing subproblems, respectively. This is fundamentally different from the setting in Muter et al. (2013a) and in this paper, which requires us to generate both columns and their associated linking constraints interdependently on-the-fly.

In this paper, we extend our previous work on CDR-problems and solve them by using Benders decomposition (Benders, 1962). This decomposition technique partitions the variables into two smaller problems –the Benders master problem and the Benders subproblem– so that the overall problem can be handled efficiently with an iterative algorithm known as delayed cut generation. Rahmaniani et al. (2016) provides a comprehensive survey of applications solved by Benders decomposition, including capacitated facility location (Fischetti et al., 2016), production routing (Adulyasak et al., 2015), multi-period hub location (Gelareh et al., 2015), and strip packing (Côté et al., 2014). As we shall discuss in the next section, the structure of CDR-problems qualifies for such a decomposition. However, we observe that a direct application of Benders decomposition is not possible because we do not have the complete description of the dual of the Benders subproblem – the dual slave problem – during the iterations. In particular, the dimension of the feasible region of the dual slave problem – the dual polyhedron – increases as new linking constraints are introduced. This novel structure leads us to reconsider the fundamental parts of Benders decomposition; solving a sequence of relaxed Benders master problems and applying delayed cut generation by solving the dual slave problem in between. The proposed analysis along with our observations constitute the main contributions of this work: We develop a new Benders decomposition methodology for solving large-scale linear programs with column-dependent-rows. This approach induces a novel integration of the delayed (Benders) cut generation and simultaneous column-and-row generation for solving large-scale CDR-problems. To illustrate the application of the proposed method, we explain

each step on the time-constrained routing problem. Our numerical experiments confirm that the new decomposition method outperforms not only the off-the-shelf solvers but also our previous algorithm for CDR-problems presented in Muter et al. (2013a).

2. Motivation. We devote this section to explaining our motivation for developing a new Benders decomposition methodology for solving CDR-problems. To this end, we first revisit the generic mathematical model for CDR-problems:

$$
\begin{array}{llll}
(\mathbf{MP}) & \text{minimize} & \sum_{k\in K} c_k y_k + \sum_{n\in N} d_n x_n, & \\
 & \text{subject to} & \sum_{k\in K} A_{jk} y_k \geq a_j, \quad j \in J, & \text{(MP-y)} \\
 & & \sum_{n\in N} B_{mn} x_n \geq b_m, \quad m \in M, & \text{(MP-x)} \\
 & & \sum_{k\in K} C_{ik} y_k + \sum_{n\in N} D_{in} x_n \geq r_i, \quad i \in I(K,N), & \text{(MP-yx)} \\
 & & y_k \geq 0, k \in K, \quad x_n \geq 0, n \in N. &
\end{array}
$$

In general, we allow for exponentially many y- and x- variables in the *master problem* formulation (**MP**) above and therefore reckon that any generic viable algorithm for solving CDR-problems must be able to generate both types of variables dynamically in a column generation framework. The cardinality of the constraints (MP-y) and (MP-x) is polynomially bounded in the size of the problem, and these are directly incorporated into the model. However, the cardinality of the set of linking constraints (MP-yx) depends on $\mid K \mid$ and $\mid N \mid$ and is either theoretically or practically too large – see Example 3.1 in Section 3. Note that the dependence of the linking constraints on the variables is conveyed through the notation $I(K,N)$. This non-standard structure of (**MP**) prompts us to search for alternatives to handle the linking constraints (MP-yx) in a column generation scheme.

A key observation that motivates the solution approach for CDR-problems in this paper is that the constraints (MP-y) and (MP-x) impose conditions only on the $y-$ and $x-$variables, respectively. Therefore, these two groups of variables can be handled in two separate problems. This is a typical structure amenable to Benders decomposition. In particular, we project out the $x-$variables and relegate them to the subproblem while the $y-$variables are kept in the master problem. This choice is not arbitrary; the underlying reason will be clear in Section 3.1, where we qualify the $y-$variables as the *primary* set of variables in some sense. Formally, we write

$$
\begin{array}{lll}
\text{minimize} & \sum_{k\in K} c_k y_k + z(\boldsymbol{y}), & \\
\text{subject to} & \sum_{k\in K} A_{jk} y_k \geq a_j, & j \in J, \\
 & y_k \geq 0, & k \in K,
\end{array}
\tag{1}
$$

where for a fixed $\bar{\boldsymbol{y}}$ we have

$$
\begin{array}{llll}
(\mathbf{BSP}) & z(\bar{\boldsymbol{y}}) = \text{minimize} & \sum_{n\in N} d_n x_n, & \\
 & (v_m) \quad \text{subject to} & \sum_{n\in N} B_{mn} x_n \geq b_m, & m \in M, \\
 & (w_i) & \sum_{n\in N} D_{in} x_n \geq r_i - \sum_{k\in K} C_{ik} \bar{y}_k, & i \in I(K,N), \\
 & & x_n \geq 0, & n \in N.
\end{array}
\tag{2}
$$

This problem is referred to as the *Benders subproblem*, where the dual variables are indicated in parentheses to the left of their respective constraints. Using LP duality, we obtain the equivalent *dual slave* problem

$$
\begin{array}{llll}
(\mathbf{DBSP}) \quad z(\bar{\boldsymbol{y}}) = & \text{maximize} & \sum_{m\in M} b_m v_m + \sum_{i\in I(K,N)} (r_i - \sum_{k\in K} C_{ik}\bar{y}_k) w_i, & \\
& \text{subject to} & \sum_{m\in M} B_{mn} v_m + \sum_{i\in I(K,N)} D_{in} w_i \le d_n, & n \in N, \\
& & v_m \ge 0, m \in M; w_i \ge 0, i \in I(K,N). &
\end{array} \tag{3}
$$

Note that the dual polyhedron, denoted by Φ, is independent of $\bar{\boldsymbol{y}}$. We assume that Φ is nonempty, and by representing the set of extreme points and extreme rays of Φ as P_Φ and Q_Φ, respectively, (**DBSP**) is re-formulated as

$$
\begin{array}{llll}
z(\bar{\boldsymbol{y}}) = & \text{minimize} & z, & \\
& \text{subject to} & \sum_{m\in M} b_m v_m + \sum_{i\in I(K,N)} (r_i - \sum_{k\in K} C_{ik}\bar{y}_k) w_i \le z, & (\boldsymbol{v},\boldsymbol{w}) \in P_\Phi, \\
& & \sum_{m\in M} b_m v_m + \sum_{i\in I(K,N)} (r_i - \sum_{k\in K} C_{ik}\bar{y}_k) w_i \le 0, & (\boldsymbol{v},\boldsymbol{w}) \in Q_\Phi.
\end{array} \tag{4}
$$

If we plug $z(\boldsymbol{y})$ into (1), then we obtain the *Benders master problem* given by

$$
\begin{array}{llll}
(\mathbf{BMP}) & \text{minimize} & \sum_{k\in K} c_k y_k + z, & \\
& \text{subject to} & \sum_{k\in K} A_{jk} y_k \ge a_j, & j \in J, \\
& & z + \sum_{i\in I(K,N)} \sum_{k\in K} C_{ik} w_i y_k \ge \sum_{m\in M} b_m v_m + \sum_{i\in I(K,N)} r_i w_i, & (\boldsymbol{v},\boldsymbol{w}) \in P_\Phi, \\
& & \sum_{i\in I(K,N)} \sum_{k\in K} C_{ik} w_i y_k \ge \sum_{m\in M} b_m v_m + \sum_{i\in I(K,N)} r_i w_i, & (\boldsymbol{v},\boldsymbol{w}) \in Q_\Phi, \\
& & y_k \ge 0, k \in K, &
\end{array} \tag{5}
$$

which is equivalent to (**MP**). On the one hand, (**BMP**) has far fewer variables compared to (**MP**) as the $x-$variables do not exist in this problem. On the other hand, the number of constraints in (**BMP**) may be substantially larger because it includes one constraint for each extreme point and extreme ray of Φ. The main motivation for conducting this study is to investigate whether the application of Benders decomposition to solve CDR-problems is advantageous given that the overall problem needs to be solved by column generation as alluded to at the start of this section.

There exist other studies in the literature, such as Cordeau et al. (2000) and Cordeau et al. (2001), which employ column generation within a Benders framework. In all of these works, the dual polyhedron is available for enumeration or generation. This is a key point because the validity of Benders decomposition rests on the independence of the dual polyhedron from the values of the variables kept in the Benders master problem. These variables only appear in the objective function of the dual slave problem, and consequently, the convergence of the Benders decomposition scheme is established by arguing the finiteness of the set of extreme points and the set of extreme rays of the dual polyhedron. Clearly, the maximum number of generated Benders cuts cannot exceed the total number of extreme points and extreme rays of the dual polyhedron. However, for CDR-problems considered here, the complete description of the dual polyhedron is not available. Note that the sets of constraints and variables in the dual slave problem are determined by the set of $x-$variables and the set of linking constraints, respectively. Consequently, if we adopt an iterative approach that generates both variables and linking constraints on the fly in order to solve (**BMP**), then the dual slave problem would be missing both constraints and variables during

the iterations of such an algorithm. In other words, we would be solving the dual slave problem over *different* feasible regions over the course of the iterations. This undermines the basic premise of Benders decomposition. Moreover, generating a set of $y-$ and $x-$ variables and an associated set of linking constraints simultaneously as a result of solving a single pricing subproblem (PSP) is a challenging task, not least because the $y-$variables are generated in the Benders master problem, while the $x-$variables and the linking constraints are added to the dual slave problem.

Based on the preceding discussion, our main goal is to design a PSP for the column generation applied to the Benders master problem which takes the changes in the dual polyhedron into consideration. We show that the generation of new $y-$variables in the master problem also entails updating the existing Benders cuts by taking into account the modifications required in the dual polyhedron in an effort to ensure the convergence of the algorithm. Ultimately, we deal with an unconventional case, where the currently existing methods fall short. We tackle it by designing an algorithm that correctly incorporates simultaneous column-and-row generation into the Benders decomposition scheme.

3. Proposed Methodology. This study depends on our previous work on simultaneous column-and-row generation (Muter et al., 2013a). To make the paper self-contained, we first review our notation about CDR-problems on an illustrative example. Subsequently, we introduce the components of our proposed approach – the delayed Benders cut generation and the simultaneous column-and-row generation – and conclude this section with a formal convergence analysis of our algorithm.

3.1 Notation and Previous Work. As we mentioned in the previous section, when column generation is applied to the master problem (**MP**), the sets of columns K and N are replaced by their subsets $\bar{K}$ and $\bar{N}$, respectively. Recall that the linking constraints depend on the columns. Therefore, any restriction on the variables leads to a restriction on the constraints, and we define the index set $\bar{I} := I(\bar{K}, \bar{N}) \subseteq I(K, N)$ to designate the set of linking constraints formed by $\{y_k \mid k \in \bar{K}\}$, and $\{x_n \mid n \in \bar{N}\}$. To reflect this special structure, we refer to the model formed by $\{y_k \mid k \in \bar{K}\}$, $\{x_n \mid n \in \bar{N}\}$, (MP-y), (MP-x), and a subset of the linking constraints indexed by $\bar{I}$ as the *short restricted master problem* (**SRMP**).

Muter et al. (2013a) lay out three assumptions that characterize a generic CDR-problem (see Appendix A). Relying on these assumptions, they propose a simultaneous column-and-row generation algorithm applied directly to (**MP**). Likewise, we also use the same assumptions in our solution approach, which is based on Benders decomposition coupled with simultaneous column-and-row generation applied to (**BMP**). Roughly speaking, these assumptions designate the $y-$variables as the *primary* set of variables in the sense that generating new $x-$variables and linking constraints does not improve the objective function value of (**SRMP**) unless certain $y-$variables already exist in (**SRMP**) or are generated simultaneously. Thus, the fundamental idea behind simultaneous column-and-row generation stems from the generation of a set of $y-$variables indexed by S_K – referred to as a minimal variable set –, which also triggers the generation of a set of $x-$variables indexed by $S_N(S_K)$ and a set of linking constraints indexed by $\Delta(S_K)$. Note that both sets depend on S_K. The second and third assumptions in Appendix A imply that the variables $\{x_n \mid n \in S_N(S_K)\}$ cannot assume non-zero values until all variables $\{y_k \mid k \in S_K\}$ are generated and take positive values. Therefore, the linking constraints $\Delta(S_K)$ are redundant until

$\{y_k \mid k \in S_K\}$ are present in (**SRMP**). These concepts and our notation – summarized in Table 1 – are best explained by an illustrative example.

Table 1: Notation (Muter et al., 2013a).

S_K	the index set of a minimal variable set $\{y_l \mid l \in S_K\}$.
S_K^k	index k denotes that y_k is a member of the minimal variable set $\{y_l \mid l \in S_K^k\}$.
$S_N(S_K)$	the index set of the $x-$variables induced by $\{y_l \mid l \in S_K\}$.
$\Delta(S_K)$	the index set of the linking constraints induced by $\{y_l \mid l \in S_K\}$.
$\mathcal{P}_k$	the power set of the set composed by the index sets of the minimal variable sets containing y_k.
$\mathcal{F}_k$	a family of the index sets of the minimal variable sets of the form S_K^k; *i.e.*, $\mathcal{F}_k \in \mathcal{P}_k$.
Σ_k	$= \cup_{S_K^k \in \mathcal{F}_k} S_K^k$.

Example 3.1 *The quadratic set covering problem (QSC) is a CDR-problem that satisfies the assumptions given in Appendix A. The reader is referred to Muter et al. (2013a) for details. In this problem, the objective is to cover all items $j \in J$ by a subset of the sets $k \in K$ at minimum total cost. The cost function contains not only the individual costs for $k \in K$ but also the cross-effect costs between each pair of sets $k, l \in K$, leading to a quadratic objective function. A linearization of the objective function yields the following LP relaxation of QSC:*

$$\text{minimize} \quad \sum_{k \in K} f_{kk} y_k + \sum_{(k,l) \in \Pi, k<l} 2 f_{kl} x_{kl}, \tag{6}$$

$$\text{subject to} \quad \sum_{k \in K} A_{jk} y_k \geq 1, \qquad j \in J, \tag{7}$$

$$y_k + y_l - x_{kl} \leq 1, \qquad (k,l) \in \Pi,\ k < l, \tag{8}$$

$$y_k - x_{kl} \geq 0, \qquad (k,l) \in \Pi,\ k < l, \tag{9}$$

$$y_l - x_{kl} \geq 0, \qquad (k,l) \in \Pi,\ k < l, \tag{10}$$

$$y_k \geq 0, \qquad k \in K, \tag{11}$$

$$x_{kl} \geq 0, \qquad (k,l) \in \Pi,\ k < l, \tag{12}$$

*where $\Pi := K \times K$ is the set of all possible pairs, and $A_{jk} = 1$, if item j is covered by set k; and is 0, otherwise. In this linearized formulation of QSC, each pair of variables induces three linking constraints, and the size of the formulation grows very quickly with $\mid K \mid$. Thus, (6)-(12) lends itself nicely to our solution methods tailored to CDR-problems. To construct (**SRMP**), a subset of the columns from K, denoted by $\bar{K}$, is selected. This induces a set of associated linking constraints indexed by $\bar{\Pi} \subset \Pi$ containing the restricted set of column pairs in $\bar{K} \times \bar{K}$ along with the related $x-$variables. The addition of a new variable y_k to (**SRMP**) introduces a set of linking constraints and $x-$variables for each pair (k,l) with $l \in \bar{K}$. Observe that before both y_k and y_l enter (**SRMP**), the variable x_{kl} and the linking constraints (8)-(10) are redundant. Thus, the pair of variables $\{y_k, y_l\}$ constitute a minimal variable set $S_K^k = \{k,l\}$ which triggers the generation of x_{kl} with $S_N(S_K^k) = \{(k,l)\}$ and the three associated linking constraints labeled as $\Delta(S_K^k)$. The challenging aspect of solving (6)-(12) via simultaneous column-and-row generation is that*

the reduced cost of y_k may only drop below zero if it is added to (**SRMP**) *along with, say, y_l, y_m, while y_n is already included in* (**SRMP**)*. The notation $\mathcal{F}_k$ and $\mathcal{P}_k$ is required to account for this possibility. In the case under consideration, we have a family of index sets $\mathcal{F}_k = \{\{k,l\},\{k,m\},\{k,n\}\}$, where each element corresponds to a minimal variable set containing y_k. Thus, the members of $\{y_k \mid k \in \Sigma_k\}$ currently missing from* (**SRMP**) *are incorporated, where $\Sigma_k = \cup_{S_K^k \in \mathcal{F}_k} S_K^k = \{k,l,m,n\}$. Furthermore, for each $S_K^k \in \mathcal{F}_k$ the associated $x-$variables indexed by $S_N(S_K^k)$ and the set of linking constraints $\Delta(S_K^k)$ are also added. Consequently, the updated* (**SRMP**) *is formed by $\{y_k \mid k \in \bar{K} \cup \Sigma_k\}$, $\{x_n \mid n \in \bar{N} \cup S_N(\Sigma_k)\}$, and the set of linking constraints $\bar{I} \cup \Delta(\Sigma_k)$ in addition to constraints* (7)*. For QSC, $\mathcal{F}_k$ will generally include several minimal sets resulting from the interaction of y_k with multiple $y-$variables. In contrast, for problems that do not posses such a structure, such as the time-constrained routing problem that we consider in Section 4, $\mathcal{F}_k$ is always a singleton.*

Starting from the concepts and observations discussed above, Muter et al. (2013a) design a simultaneous column-and-row generation algorithm for solving (**MP**) to optimality. This algorithm invokes three PSPs sequentially and terminates if no new column is generated in a full cycle. In the $y-$ and $x-$PSPs, the currently absent $y-$ and $x-$ variables are priced out, respectively, by ignoring the missing linking constraints. In the row-generating PSP, the objective is to determine a family $\mathcal{F}_k$ of index sets S_K^k that minimizes the reduced cost of y_k. If this PSP identifies a $y-$variable with a negative reduced cost, then (**SRMP**) grows both horizontally and vertically as discussed in the context of QSC in the preceding example. A fundamental difference of our approach in this paper from that in Muter et al. (2013a) is that we exploit the decomposable structure of (**MP**) instead of attacking it directly. This fresh perspective, on the one hand, carries computational benefits as demonstrated in Section 4, and on the other hand, poses the theoretical challenges discussed at the end of Section 2.

3.2 Delayed Cut Generation. In this section, we restrict our attention to delayed Benders cut generation for solving (**BMP**). In order not to detract the reader's attention from this focus, we assume that all $y-$variables are present in (**BMP**) and tackle the integration of delayed Benders cut generation and simultaneous column-and-row generation in the next section.

Clearly, generating the complete set of Benders cuts in (**BMP**) by enumerating all extreme points and extreme rays of the dual polyhedron is computationally prohibitive. A well-known remedy is to employ delayed cut generation, which starts out with a small number of Benders cuts present in the Benders master problem – referred to as the short Benders master problem (**S-BMP**) – and augments the set of Benders cuts iteratively by solving a subproblem. Each iteration involves the re-optimization of (**S-BMP**) followed by Benders cut generation based on the solution of (**DBSP**). The sets of extreme points and extreme rays of the dual polyhedron for which cuts have been incorporated into (**S-BMP**) over $t = 0, 1, \ldots$ iterations are represented by $P_\Phi^t \subset P_\Phi$ and $Q_\Phi^t \subset Q_\Phi$, respectively. The optimal objective function value and the corresponding optimal solution of (**S-BMP**) at iteration t are denoted by Z^t and $(\bar{\boldsymbol{y}}^t, \bar{z}^t)$, respectively. At any iteration t, the value of Z^t is a lower bound on the optimal objective function value of (**BMP**) because (**S-BMP**) is a relaxation of (**BMP**). The relaxation gets tighter as we keep adding Benders cuts over the iterations; *i.e.*, we have $P_\Phi^{t-1} \subseteq P_\Phi^t$ and $Q_\Phi^{t-1} \subseteq Q_\Phi^t$ for $t = 1, \cdots$. Thus, the sequence of objective function values Z^t, $t = 0, 1. \ldots$, is non-decreasing.

At each iteration, (**DBSP**) is solved for the current optimal solution of (**S-BMP**). If (**DBSP**) is unbounded – that is, if (**BSP**) is infeasible –, we retrieve the corresponding extreme ray $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$, set $Q_\Phi^{t+1} = Q_\Phi^t \cup \{(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})\}$, and add the following feasibility cut to (**S-BMP**):

$$\sum_{i \in I(K,N)} \sum_{k \in K} C_{ik} \bar{w}_i y_k \geq \sum_{m \in M} b_m \bar{v}_m + \sum_{i \in I(K,N)} r_i \bar{w}_i. \tag{FeasCut}$$

Otherwise, there exists an extreme point optimal solution $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ to (**DBSP**). If $z(\bar{\boldsymbol{y}}^t) > \bar{z}^t$ holds for this extreme point, then we set $P_\Phi^{t+1} = P_\Phi^t \cup \{(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})\}$ and incorporate the following violated optimality cut into (**S-BMP**):

$$z + \sum_{i \in I(K,N)} \sum_{k \in K} C_{ik} \bar{w}_i y_k \geq \sum_{m \in M} b_m \bar{v}_m + \sum_{i \in I(K,N)} r_i \bar{w}_i. \tag{OptCut}$$

Finally, if (**DBSP**) is not unbounded and no optimality cut is violated, the algorithm terminates. This stopping condition may be too conservative, and we may also terminate the delayed cut generation algorithm if we are sufficiently close to optimality. Observe that summing up the optimal objective function value $z(\bar{\boldsymbol{y}}^t)$ of (**DBSP**) with $\sum_{k \in K} c_k \bar{y}_k^t$ yields an upper bound on the optimal objective function value of (**BMP**) at iteration t. However, the sequence of these values is not necessarily non-increasing. Therefore, we need to keep track of the smallest upper bound (incumbent upper bound) identified so far. A natural termination condition for the algorithm is then that the gap between Z^t and the incumbent upper bound drops below a predetermined small value.

3.3 Simultaneous Column-and-Row Generation. We now propose an iterative column-and-row generation algorithm in combination with delayed Benders cut generation to solve CDR-problems to optimality. Each iteration of the algorithm involves solving a Benders master problem followed by a series of subproblems in some appropriate sequence –as discussed later in this section– in order to generate new variables, linking constraints, and Benders cuts. Initially, we start out by replacing the index set K of the set of $y-$variables in (**S-BMP**) by a subset $\bar{K}$. Due the structure of CDR-problems, the $y-$variables indexed by $\bar{K}$ induce a specific set of $x-$variables indexed by $\bar{N} \subset N$ and a set of linking constraints $\bar{I} \subset I(K,N)$. Consequently, both variables and constraints are absent from the Benders subproblem and the dual slave problem during the iterations. We aptly refer to these problems as the short-and-restricted Benders subproblem (**SR-BSP**) and the short-and-restricted dual slave problem (**SR-DBSP**) and state them below for a given $\bar{\boldsymbol{y}}^t$ at iteration t:

$$\begin{array}{llll} (\textbf{SR-BSP}) \quad z(\bar{\boldsymbol{y}}^t) = & \text{minimize} & \sum_{n \in \bar{N}} d_n x_n, & \\ & \text{subject to} & \sum_{n \in \bar{N}} B_{mn} x_n \geq b_m, & m \in M, \\ & & \sum_{n \in \bar{N}} D_{in} x_n \geq r_i - \sum_{k \in \bar{K}} C_{ik} \bar{y}_k^t, & i \in \bar{I}, \\ & & x_n \geq 0, & n \in \bar{N}; \end{array} \tag{13}$$

$$\begin{array}{llll} (\textbf{SR-DBSP}) \quad z(\bar{\boldsymbol{y}}^t) = & \text{maximize} & \sum_{m \in M} b_m v_m + \sum_{i \in \bar{I}} (r_i - \sum_{k \in \bar{K}} C_{ik} \bar{y}_k^t) w_i, & \\ & \text{subject to} & \sum_{m \in M} B_{mn} v_m + \sum_{i \in \bar{I}} D_{in} w_i \leq d_n, & n \in \bar{N}, \\ & & v_m \geq 0, m \in M; w_i \geq 0, i \in \bar{I}. & \end{array} \tag{14}$$

A fundamental point that deserves special attention is the evolution of the dual polyhedron over the iterations. Both the set of variables w_i, $i \in \bar{I}$, and the set of constraints $n \in \bar{N}$ in (**SR-DBSP**) are

determined by $y_k,\ k \in \bar{K}$, kept in the short-and-restricted Benders master problem (**SR-BMP**) given in the following. The implication is that both dimensions (variables) and defining inequalities (constraints) are initially removed from the dual polyhedron Φ, which are added back as required at certain steps of the algorithm. We label the modified dual polyhedron as $\bar{\Phi}$ and index it by s. Under this definition, (**SR-BMP**) at iteration t is formulated as

$$
\begin{array}{llll}
(\textbf{SR-BMP}) & \text{minimize} & \sum_{k\in\bar{K}} c_k y_k + z, & \\
(u_j) & \text{subject to} & \sum_{k\in\bar{K}} A_{jk} y_k \geq a_j, & j \in J, \\
(\gamma_p) & & z + \sum_{k\in\bar{K}} \sum_{i\in\bar{I}} C_{ik} w_i y_k \geq \sum_{m\in M} b_m v_m + \sum_{i\in\bar{I}} r_i w_i, & (\boldsymbol{v},\boldsymbol{w}) \in P^t_{\bar{\Phi}^s}, \\
(\mu_q) & & \sum_{k\in\bar{K}} \sum_{i\in\bar{I}} C_{ik} w_i y_k \geq \sum_{m\in M} b_m v_m + \sum_{i\in\bar{I}} r_i w_i, & (\boldsymbol{v},\boldsymbol{w}) \in Q^t_{\bar{\Phi}^s}, \\
 & & y_k \geq 0, & k \in \bar{K}.
\end{array}
\tag{15}
$$

Notation. For brevity of presentation, going forward we assume that the extreme points and extreme rays of $\bar{\Phi}^s$ are enumerated sequentially and substitute the notation $p \in P^t_{\bar{\Phi}^s}$ for $(\boldsymbol{v},\boldsymbol{w}) \in P^t_{\bar{\Phi}^s}$ and $q \in Q^t_{\bar{\Phi}^s}$ for $(\boldsymbol{v},\boldsymbol{w}) \in Q^t_{\bar{\Phi}^s}$. Moreover, $(\boldsymbol{v}^p,\boldsymbol{w}^p)$ stands for the pth extreme point or ray in $P^t_{\bar{\Phi}^s}$ or $Q^t_{\bar{\Phi}^s}$, respectively, depending on the context. Also note that the index s in $\bar{\Phi}^s$ is independent from the iteration index t as we explain in the discussion following Figure 1 below.

As we elaborated upon in Section 2, the invariance of the dual polyhedron over the iterations is a basic pillar of Benders decomposition. Therefore, we must ensure that all previously generated Benders cuts remain valid with respect to the original dual polyhedron Φ as $\bar{\Phi}$ is updated. This is a main concern in the PSPs explained later in this section. Note that we cannot trivially argue the validity of the previously generated cuts because the relation $\bar{\Phi}^0 \subseteq \bar{\Phi}^1 \subseteq \bar{\Phi}^2 \subseteq \ldots \subseteq \Phi$ does not necessarily hold when both variables and constraints are incorporated back into the dual polyhedron iteratively. Nonetheless, as we later show, the extreme points and extreme rays of the lower-dimensional dual polyhedron $\bar{\Phi}^s$ can be augmented with additional variables so that they remain extreme points and extreme rays for the higher-dimensional –but more constrained– dual polyhedron $\bar{\Phi}^{s+1}$. At termination, all Benders cuts in (**SR-BMP**) are valid with respect to Φ – see Theorem 3.1.

We now discuss the rationale of the proposed algorithm at a high level without going into the details of any PSP. An overview of the proposed column-and-row generation algorithm is depicted in Figure 1. Each iteration –indexed by t– commences by solving the current (**SR-BMP**) via column generation to optimality by calling the $y-$PSP repeatedly and augmenting (**SR-BMP**) with new $y-$variables as required. At the conclusion of this column generation phase, we identify the minimal variables sets that were previously missing from (**SR-BMP**) and are completed incidentally by the addition of new $y-$variables. Recall that by Assumption A.2, a linking constraint is redundant until all variables in at least one of the associated minimal variable sets are generated. The Benders subproblem is updated by taking into account the linking constraints induced by these new minimal variable sets. In some CDR-problems, such as the quadratic set covering problem given in Example 3.1, the $x-$variables which appear in these new linking constraints are readily available and also incorporated directly. At this point, we can correctly evaluate whether the current optimal solution of (**SR-BMP**) yields a feasible solution in the Benders subproblem by verifying the boundedness of (**SR-DBSP**). If not, the current optimal solution of (**SR-BMP**) is removed from further consideration by generating a Benders feasibility cut (FeasCut)

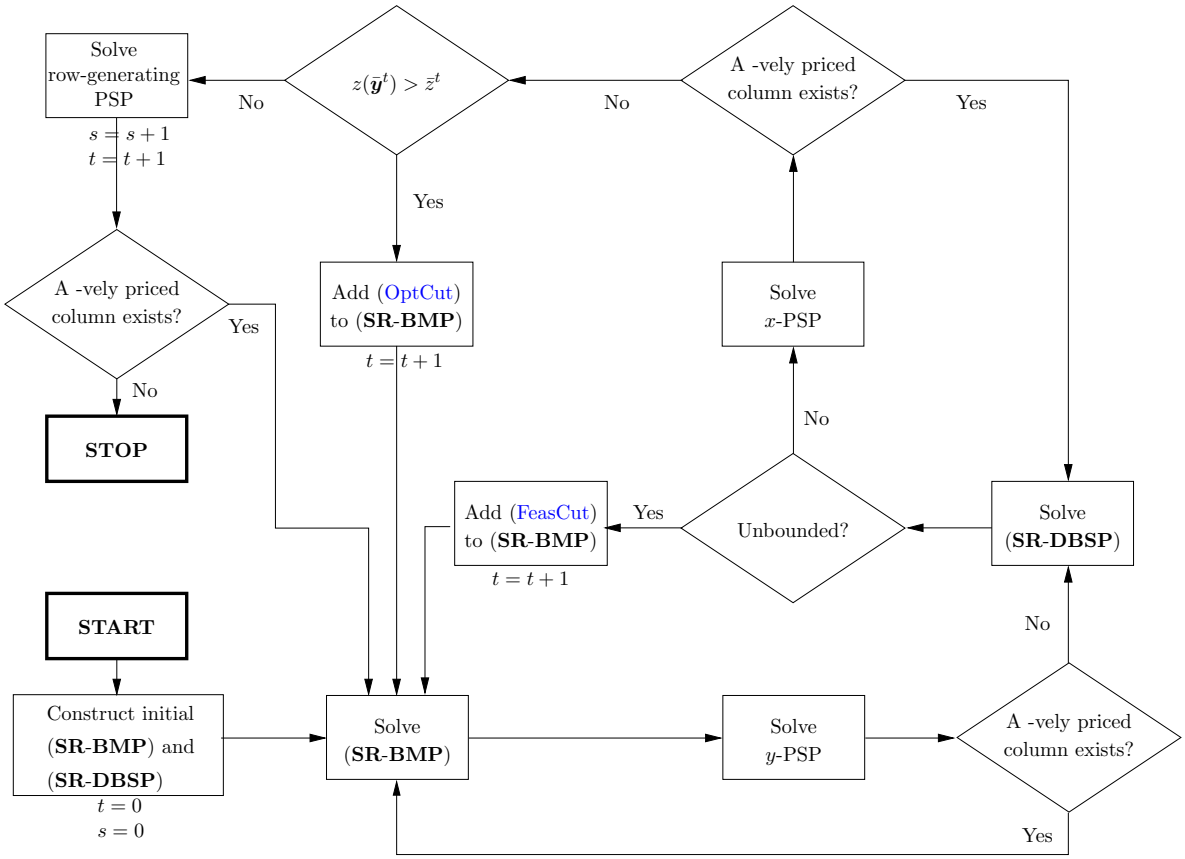

Figure 1: The flow chart of the proposed algorithm.

based on an extreme ray of (**SR-DBSP**), and column generation resumes for (**SR-BMP**). Once we establish that the optimal solution of the current (**SR-BMP**) is feasible with respect to the Benders subproblem, we move on to improving the approximation of the objective function of (**SR-BMP**). To this end, we first solve (**SR-BSP**) to optimality through column generation by invoking the $x-$PSP repeatedly with the current set of available linking constraints. Observe that the feasibility of (**SR-BSP**) is guaranteed at this step because (**SR-DBSP**) is bounded. At the completion of the column generation, a new optimality cut (OptCut) created from an extreme point optimal solution of (**SR-BSP**) is appended to (**SR-BMP**) if a violated cut is detected as discussed in Section 3.2. Otherwise, the row-generating PSP is called. (**BMP**) is solved to optimality and our column-and-row generation algorithm in combination with delayed Benders cut generation terminates whenever the row-generating PSP fails to identify a $y-$variable with a negative reduced cost.

We underline a fundamental difference between the row-generating PSP and the other two, before we delve into the specifics of these subproblems. The $y-$ and $x-$PSPs generate new $y-$ and $x-$variables, respectively, by completely ignoring the missing linking constraints. That is, these two PSPs do not change the dimension of the dual polyhedron $\bar{\Phi}$. In contrast, the whole purpose of the row-generating PSP is to determine at least one $y-$variable which prices out favorably, only if it is created along with other $y-$ and $x-$variables and new linking constraints. The row-generating PSP triggers the addition of a set of defining constraints and dimensions to the dual polyhedron $\bar{\Phi}^s$ and increases its index from s to $s + 1$. In other words, the index s in $\bar{\Phi}^s$ keeps track of the number of times the dimension of the

dual polyhedron grows and the Benders cuts present in (**SR-BMP**) are modified accordingly. We do not indicate it in Figure 1; however, strictly speaking, s is also incremented when the necessary changes to the dual polyhedron are applied after the $y-$PSP as a consequence of the incidentally completed minimal variable sets. We also point out that the algorithmic flow in Figure 1 avoids solving the computationally more expensive row-generating PSP as long as it can proceed otherwise.

***y*−Pricing Subproblem.** This subproblem is handled in (**SR-BMP**), and the objective is to determine a variable y_k, $k \in (K \setminus \bar{K})$ with a negative reduced cost, where any impact on the dual polyhedron is disregarded. The $y-$PSP is defined as

$$\zeta_y = \min_{k \in K} \left\{ c_k - \sum_{j \in J} u_j A_{jk} - \sum_{p \in P^t_{\bar{\Phi}^s}} \sum_{i \in \bar{I}} C_{ik} w_i^p \gamma_p - \sum_{q \in Q^t_{\bar{\Phi}^s}} \sum_{i \in \bar{I}} C_{ik} w_i^q \mu_q \right\}, \tag{16}$$

where $u_j \geq 0, j \in J$, $\gamma_p \geq 0, p \in P^t_{\bar{\Phi}^s}$, and $\mu_q \geq 0, q \in Q^t_{\bar{\Phi}^s}$ are the set of dual variables corresponding to the constraints in (**SR-BMP**) with their values retrieved from the optimal solution of the current (**SR-BMP**). Note that the values of the dual variables corresponding to the Benders cuts present in (**SR-BMP**) are taken into account in this subproblem. If ζ_y is non-negative, then the current solution $\bar{\boldsymbol{y}}^t$ of (**SR-BMP**) is proven optimal, and we proceed to solve (**SR-DBSP**) given $\bar{\boldsymbol{y}}^t$. Otherwise, there exists y_k whose reduced cost, denoted by $\bar{c}_k$, is negative, and (**SR-BMP**) grows by a single variable by setting $\bar{K} \leftarrow \bar{K} \cup \{k\}$.

The $y-$PSP is not intended to generate new linking constraints. However, the $y-$variables added by the $y-$PSP may incidentally constitute or complete new minimal variable sets in (**SR-BMP**) that induce a new set of linking constraints and associated $x-$variables in (**SR-BSP**). These new linking constraints and $x-$variables change the dual polyhedron, and – as alluded to before – prescribe that we update the currently existing Benders cuts in (**SR-BMP**) to ensure the correctness of our algorithm. Moreover, as explained previously in this section, the mandatory modifications must be reflected in the dual polyhedron prior to solving (**SR-DBSP**) so that the feasibility of the current optimal solution $\bar{\boldsymbol{y}}^t$ of (**SR-BMP**) is assessed properly. Therefore, after generating columns by solving the $y-$PSP the incidentally completed minimal variable sets must be identified, and the associated changes must be applied to (**SR-BSP**), (**SR-DBSP**), and to the existing Benders cuts in (**SR-BMP**). The details of the cut update procedure are discussed in the context of the row-generating PSP in the following.

***x*−Pricing Subproblem.** This subproblem is part of the column generation algorithm for solving (**SR-BSP**) to optimality by generating new $x-$variables, assuming that the set of linking constraints stays fixed. According to Assumption A.1 in Appendix A, no new linking constraint may be induced without generating new $y-$variables; that is, $\Delta(\emptyset) = \emptyset$ for this PSP. The $x-$PSP is then stated as

$$\zeta_x = \min_{n \in N_{\bar{K}}} \left\{ d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in \bar{I}} D_{in} w_i \right\}, \tag{17}$$

where $N_{\bar{K}}$ is the set of $x-$variables induced by $\bar{K}$. If $\zeta_x < 0$, we set $\bar{N} \leftarrow \bar{N} \cup \{n\}$, and (**SR-BSP**) and (**SR-DBSP**) grow by a single variable and constraint, respectively. Otherwise, we check whether the relation (OptCut) is violated by the optimal solution of (**SR-DBSP**) and introduce a new optimality cut into (**SR-BMP**) if necessary. Observe that even if the $x-$PSP creates new constraints in (**SR-DBSP**) between two consecutively generated optimality cuts, the former cut stays valid as long as $N_{\bar{K}}$ stays

constant. This is true because the optimal extreme point solution of (**SR-DBSP**) does not violate any constraint in $n \in N_{\bar{K}} \setminus \bar{N}$. Recall that $N_{\bar{K}}$ is only augmented with additional $x-$variables as a consequence of the row-generating PSP or if some minimal variable sets are completed incidentally in the $y-$PSP.

Row-Generating Pricing Subproblem and Lifting. We now develop the row-generating PSP which identifies the $y-$variables that price out favorably only if they are incorporated into (**SR-BMP**) along with new linking constraints and $x-$variables. That is, differently from the $y-$PSP, this PSP purposely searches for minimal variable sets to be added to (**SR-BMP**) so that the reduced cost of at least one $y-$variable is decreased below zero. Consequently, the outcome of the row-generating PSP is the $y-$variable with the smallest reduced cost, say y_k, and the associated sets $\mathcal{F}_k$, Σ_k, $S_N(\Sigma_k)$ and $\Delta(\Sigma_k)$ as defined in Section 3.1. On the one hand, the new set of $y-$variables $\{y_l \mid l \in \Sigma_k\}$ are appended to (**SR-BMP**) and complete the minimal variable sets $S_K^k \in \mathcal{F}_k$. On the other hand, (**SR-BSP**) is augmented with the new linking constraints $i \in \Delta(\Sigma_k)$ featuring the new $x-$variables $\{x_n \mid n \in S_N(\Sigma_k)\}$. In other words, the column and row generation aspects of our simultaneous column-and-row generation algorithm are handled in two different problems (**SR-BMP**) and (**SR-BSP**), respectively. This separation complicates the column-and-row generation algorithm and has critical implications for the dual polyhedron. Each new variable $x_n, n \in S_N(\Sigma_k)$ introduces a new constraint into (**SR-DBSP**), and each new linking constraint $i \in \Delta(\Sigma_k)$ increases the dimension of the dual polyhedron $\bar{\Phi}^s$ in the current iteration by adding a new variable $w_i, i \in \Delta(\Sigma_k)$. From the perspective of solving (**BMP**) to optimality via generating $y-$variables and Benders cuts, two issues are rooted in these changes in the dual polyhedron. First, for reasons detailed in Section 2 and at the start of this section, we must guarantee that each Benders cut in (**SR-BMP**) corresponds to either an extreme point or an extreme ray of the updated dual polyhedron $\bar{\Phi}^{s+1}$ to ensure the convergence of our algorithm. Second, the new variables $w_i, i \in \Delta(\Sigma_k)$ appear in the existing Benders cuts constructed from $p \in P^t_{\bar{\Phi}^s}$ and $q \in Q^t_{\bar{\Phi}^s}$, and their unknown values factor into the reduced costs of the $y-$variables. A detailed analysis of both of these issues below culminates in the formulation of the row-generating PSP.

From $\bar{\Phi}^s$ to $\bar{\Phi}^{s+1}$, the dimension increases by $|\Delta(S_K^k)|$ for each $S_K^k \in \mathcal{F}_k$ added to (**SR-BMP**), and these new variables $w_i, i \in \Delta(S_K^k)$ for $S_K^k \in \mathcal{F}_k$ must also be incorporated into the *existing* Benders cuts. For instance, if we consider the optimality cut for $p \in P^t_{\bar{\Phi}^s}$, then we observe

$$z + \underbrace{\sum_{l \in \bar{K} \cup \Sigma_k} \sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} C_{il} w_i y_l}_{\text{updated term}} \geq \sum_{m \in M} b_m v_m + \sum_{i \in \bar{I}} r_i w_i + \underbrace{\sum_{i \in \Delta(\Sigma_k)} r_i w_i}_{\text{new term}}. \tag{18}$$

Initially, setting $w_i = 0, i \in \Delta(\Sigma_k)$ seems convenient. However, the validity of this approach would hinge upon whether $(\boldsymbol{v}^p, \boldsymbol{w}^p)$ remains feasible with respect to $\bar{\Phi}^{s+1}$ even if it is not an extreme point any more. This is not necessarily the case, and $(\boldsymbol{v}^p, \boldsymbol{w}^p)$ augmented with $w_i = 0, i \in \Delta(\Sigma_k)$ may violate some of the new constraints in (**SR-DBSP**) associated with the new variables $\{x_n \mid n \in S_N(\Sigma_k)\}$:

$$\sum_{m \in M} B_{mn} v_m + \sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} D_{in} w_i = \sum_{m \in M} B_{mn} v_m + \sum_{i \in \Delta(S_K^k)} D_{in} w_i \leq d_n, \; n \in S_N(S_K^k), S_K^k \in \mathcal{F}_k. \tag{19}$$

The equality on the left hand side of (19) is a consequence of Assumption A.1 in Appendix A which partitions the $x-$variables into disjoint sets of linking constraints. The new $x-$variables $x_n, n \in S_N(S_K^k), S_K^k \in \mathcal{F}_k$ are not part of the existing linking constraints $i \in \bar{I}$ in (**SR-BSP**). Consequently,

for each $n \in S_N(S_K^k), S_K^k \in \mathcal{F}_k$, the corresponding constraint in (**SR-DBSP**) involves only the variables $w_i, i \in \Delta(S_K^k)$. Ultimately, if $(\boldsymbol{v}^p, \boldsymbol{w}^p)$ augmented with $w_i = 0, i \in \Delta(\Sigma_k)$ does not satisfy any one of the constraints (19), then the corresponding Benders cut in the current (**SR-BMP**) is invalid with respect to $\bar{\Phi}^{s+1}$. The remedy is to *lift* the variables $w_i, i \in \Delta(\Sigma_k)$ in such a way that we obtain an extreme, and hence valid, point of $\bar{\Phi}^{s+1}$ out of $p \in P^t_{\bar{\Phi}^s}$ and then update the associated Benders cut in (**SR-BMP**) accordingly. Clearly, an analogous argument applies to the extreme rays $q \in Q^t_{\bar{\Phi}^s}$ and the corresponding feasibility cuts in the current (**SR-BMP**). Lifting is carried out independently for each $p \in P^t_{\bar{\Phi}^s}$ and $q \in Q^t_{\bar{\Phi}^s}$, where the values of the new dual variables $w_i, i \in \Delta(\Sigma_k)$ for p and q after lifting are represented by $w_i^p, i \in \Delta(\Sigma_k)$ and $w_i^q, i \in \Delta(\Sigma_k)$, respectively. The lifting problems are embedded into the definition of the row-generating PSP because the values $w_i^p, i \in \Delta(\Sigma_k), p \in P^t_{\bar{\Phi}^s}$ and $w_i^q, i \in \Delta(\Sigma_k), q \in Q^t_{\bar{\Phi}^s}$ also affect the reduced costs of the $y-$variables. They must be chosen such that the Benders cuts currently present in (**SR-BMP**) are lifted correctly and the $y-$variable with the minimum reduced cost is identified. Finally, note that strictly speaking it would *not* be incorrect to transform $p \in P^t_{\bar{\Phi}^s}$ or $q \in Q^t_{\bar{\Phi}^s}$ to any feasible point or ray of $\bar{\Phi}^{s+1}$. However, we insist that each Benders cut corresponds to an extreme point or ray of $\bar{\Phi}^{s+1}$ because the convergence of a Benders algorithm is argued through the finiteness of the sets of extreme points and extreme rays of the dual polyhedron.

To complete the development of the row-generating PSP we next analyze the reduced cost of a given variable y_k. The formulation of (**SR-BMP**) reveals that the reduced cost of y_k is computed as

$$\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - \sum_{p \in P^t_{\bar{\Phi}^s}} \sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} C_{ik} w_i^p \gamma_p - \sum_{q \in Q^t_{\bar{\Phi}^s}} \sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} C_{ik} w_i^q \mu_q. \tag{20}$$

We observe that the expression (20) is similar to the reduced cost of a $y-$variable in the context of the $y-$PSP given in (16), except that (20) includes the extra term $-\sum_{p \in P^t_{\bar{\Phi}^s}} \sum_{i \in \Delta(\Sigma_k)} C_{ik} w_i^p \gamma_p - \sum_{q \in Q^t_{\bar{\Phi}^s}} \sum_{i \in \Delta(\Sigma_k)} C_{ik} w_i^q \mu_q$. The challenge here is to properly account for the values $w_i^p, i \in \Delta(\Sigma_k), p \in P^t_{\bar{\Phi}^s}$ and $w_i^q, i \in \Delta(\Sigma_k), q \in Q^t_{\bar{\Phi}^s}$, which are currently unknown. Muter et al. (2013a) face a similar hurdle and design a row-generating PSP to implicitly calculate the optimal values of the dual variables of the missing linking constraints in (**SRMP**). By adopting a thinking-ahead approach similar in spirit, it turns out that we can anticipate the true values of the new dual variables $w_i, i \in \Delta(\Sigma_k)$ for any given $p \in P^t_{\bar{\Phi}^s}$ or $q \in Q^t_{\bar{\Phi}^s}$ without explicitly solving an updated version of (**SR-DBSP**) first. This look-ahead feature allows the correct computation of the reduced costs of all $y-$variables currently absent from (**SR-BMP**).

We have imposed two conditions on the row-generating subproblem so far: the values $w_i^p, i \in \Delta(\Sigma_k), p \in P^t_{\bar{\Phi}^s}$ and $w_i^q, i \in \Delta(\Sigma_k), q \in Q^t_{\bar{\Phi}^s}$ must be set so that they minimize the reduced cost of y_k, where $k \notin \bar{K}$, and they transform $p \in P^t_{\bar{\Phi}^s}$ or $q \in Q^t_{\bar{\Phi}^s}$ into an extreme point or ray of $\bar{\Phi}^{s+1}$, respectively. However, one additional trouble must be taken into account in the design of the row-generating PSP. The values $w_i^p, i \in \Delta(\Sigma_k), p \in P^t_{\bar{\Phi}^s}$ and $w_i^q, i \in \Delta(\Sigma_k), q \in Q^t_{\bar{\Phi}^s}$ also affect the reduced costs of $y_l, l \in \bar{K}$, and we must mandate that the row-generating PSP maintains the current reduced costs for these variables. This is obviously crucial to be able to warm start the simplex method for (**SR-BMP**) following the updates prescribed by the outcome of the row-generating PSP. To this end, we enforce $C_{il} w_i^p = 0$ for $l \in \bar{K}$, $i \in \Delta(\Sigma_k), p \in P^t_{\bar{\Phi}^s}$ and $C_{il} w_i^q = 0$ for $l \in \bar{K}$, $i \in \Delta(\Sigma_k), q \in Q^t_{\bar{\Phi}^s}$. This clearly guarantees that the extra term $-\sum_{p \in P^t_{\bar{\Phi}^s}} \sum_{i \in \Delta(\Sigma_k)} C_{il} w_i^p \gamma_p - \sum_{q \in Q^t_{\bar{\Phi}^s}} \sum_{i \in \Delta(\Sigma_k)} C_{il} w_i^q \mu_q$ in (20) is zero for $y_l, l \in \bar{K}$, and $\bar{c}_l$ stays constant. The mechanism, through which these conditions are incorporated into the definition of the row-generating PSP, is based on partitioning $\Delta(S_K^k)$ into two subsets for each

$S_K^k \in \mathcal{F}_k$: $\Delta_+(S_K^k)$ is the index set of the linking constraints of the form (38) given in Assumption A.3 with $C_{ik} > 0$, and $\Delta_0(S_K^k) = \Delta(S_K^k) \backslash \Delta_+(S_K^k)$. Observe that a variable $y_l, l \in \bar{K}$ cannot appear in a new linking constraint $i \in \Delta_+(S_K^k)$, and $w_i, i \in \Delta_+(S_K^k)$ are allowed to take positive values. However, a new linking constraint $i \in \Delta_0(S_K^k)$ may involve $y_l, l \in \bar{K}$; and therefore, $w_i = 0, i \in \Delta_0(S_K^k)$ is imposed in the row-generating PSP:

$$\zeta_{yx} = \min_{k \in (K \backslash \bar{K})} \Bigg\{ c_k - \sum_{j \in J} A_{jk} u_j - \sum_{p \in P^t_{\bar{\Phi}^s}} \sum_{i \in \bar{I}} C_{ik} w_i^p \gamma_p - \sum_{q \in Q^t_{\bar{\Phi}^s}} \sum_{i \in \bar{I}} C_{ik} w_i^q \mu_q$$
$$- \max_{\mathcal{F}_k \in \mathcal{P}_k} \left(\sum_{p \in P^t_{\bar{\Phi}^s}} \gamma_p \sum_{S_K^k \in \mathcal{F}_k} \alpha_{S_K^k}(p) + \sum_{q \in Q^t_{\bar{\Phi}^s}} \mu_q \sum_{S_K^k \in \mathcal{F}_k} \alpha_{S_K^k}(q) \right) \Bigg\},$$

where the optimization problem

$$\alpha_{S_K^k}(r) = \text{maximize} \quad \sum_{i \in \Delta(S_K^k)} C_{ik} w_i^r, \tag{21a}$$

$$\text{subject to} \quad \sum_{i \in \Delta(S_K^k)} D_{in} w_i^r \leq d_n(r) - \sum_{m \in M} B_{mn} v_m^r, \quad n \in S_N(S_K^k), \tag{21b}$$

$$w_i^r = 0, \quad i \in \Delta_0(S_K^k), \tag{21c}$$

$$w_i^r \geq 0, \quad i \in \Delta_+(S_K^k), \tag{21d}$$

$$|\Delta(S_K^k)| \text{ many linearly independent tight constraints among (21b)-(21d)} \tag{21e}$$

provides us with the values $w_i^p, i \in \Delta(\Sigma_k)$ or $w_i^q, i \in \Delta(\Sigma_k)$, depending on whether it is invoked with the argument $r = p \in P^t_{\bar{\Phi}^s}$ or $r = q \in Q^t_{\bar{\Phi}^s}$, respectively. The parameter $d_n(r)$ on the right hand side of (21b) is equal to d_n if (21) is solved for $p \in P^t_{\bar{\Phi}^s}$ and is zero for $q \in Q^t_{\bar{\Phi}^s}$. The underlying reason is that extreme rays are required to be feasible with respect to the recession cone of the underlying polyhedron.

In its most general form stated above, the row-generating PSP is basically a subset selection problem, where we search for the best family of the index sets of the minimal variable sets involving y_k in an effort to minimize the reduced cost of y_k. To evaluate the contribution of a given family $\mathcal{F}_k \in \mathcal{P}_k$, we solve the optimization problem (21) for each $S_K^k \in \mathcal{F}_k$, $p \in P^t_{\bar{\Phi}^s}$, and $q \in Q^t_{\bar{\Phi}^s}$. Ultimately, we identify the best combination of $y-$variables, which – if introduced simultaneously into (**SR-BMP**) – prompt the generation of a set of new linking constraints and $x-$variables such that y_k attains its minimum reduced cost. Then, we pick the $y-$variable with the smallest reduced cost among those currently missing from (**SR-BMP**). In the next section, we prove that (**BMP**) is solved to optimality if $\zeta_{yx} \geq 0$ and no missing $y-$variable prices out favorably.

Observe that all three design requirements stipulated for the row-generating PSP are reflected in the formulation above. The values $w_i^p, i \in \Delta(\Sigma_k), p \in P^t_{\bar{\Phi}^s}$ and $w_i^q, i \in \Delta(\Sigma_k), q \in Q^t_{\bar{\Phi}^s}$ are computed via (21) by respecting the restrictions on the dual variables $w_i, i \in \Delta_+(S_K^k)$ and $w_i, i \in \Delta_0(S_K^k)$ expressed through the constraints (21c)-(21d) and such that $\bar{c}_k$ is minimized. Furthermore, recall that the solution of the row-generating PSP increases the dimension of (**SR-DBSP**) by $\mid \Delta(S_K^k) \mid$ for each $S_K^k \in \mathcal{F}_k$. Thus, transforming $p \in P^t_{\bar{\Phi}^s}$ or $q \in Q^t_{\bar{\Phi}^s}$ into an extreme point or ray of $\bar{\Phi}^{s+1}$, respectively, requires that an additional $\mid \Delta(S_K^k) \mid$ linearly independent defining inequalities of (**SR-DBSP**) are satisfied at equality. This is precisely what is enforced by the constraint (21e). At first glance, this constraint seems redundant

because one may expect that the LP (21a)-(21d) would naturally admit an extreme point optimal solution satisfying (21e) automatically whenever it is feasible. However, this LP is always unbounded as we prove next, and (21e) needs to be stated explicitly as part of the row-generating PSP. Following Lemma 3.1, we also show that the optimization problem (21) always admits an optimal solution and complete our discussion on the design and structure of the row-generating PSP.

LEMMA 3.1 *The linear program* (21a)-(21d) *is unbounded.*

PROOF. It is sufficient to argue that (21a)-(21d) does always possess a feasible solution $\mathbf{w}^r$ and then construct a ray of unboundedness $\bar{\mathbf{w}}^r$. Assumption A.3 guarantees that there always exists a linking constraint $i' \in \Delta(S_K^k)$ of the form (38) such that $C_{i'k} > 0$ and $D_{i'n} < 0$ for all $n \in S_N(S_K^k)$. This prompts us to define $\mathbf{w}^r = \begin{pmatrix} 0 & 0 & \cdots & L & \cdots & 0 \end{pmatrix}^\intercal$ and $\bar{\mathbf{w}}^r = \begin{pmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 \end{pmatrix}^\intercal$, where the non-zero entries correspond to i' and L is a positive number. By definition, $i' \in \Delta_+(S_K^k)$ and (21c)-(21d) are satisfied by $\mathbf{w}$ and $\bar{\mathbf{w}}^r$. Clearly, we can always pick L large enough so that $\sum_{i \in \Delta(S_K^k)} D_{in} w_i^r = D_{i'n} L \leq d_n - \sum_{m \in M} B_{mn} v_m^r$ for $n \in S_N(S_K^k)$. Moreover, we have $\sum_{i \in \Delta(S_K^k)} D_{in} \bar{w}_i^r = D_{i'n} < 0$ for $n \in S_N(S_K^k)$ and $\sum_{i \in \Delta(S_K^k)} C_{ik} \bar{w}_i^r = C_{i'k} > 0$ which completes the proof. □

LEMMA 3.2 *The optimization problem* (21) *always has an optimal solution.*

PROOF. The polyhedron (21b)-(21d) is not empty by Lemma 3.1 and does not contain a line because all variables are bounded through the constraints (21c)-(21d). Consequently, (21b)-(21d) has at least one extreme point (Bertsimas and Tsitsiklis, 1997, Theorem 2.6) and the extreme point with the lowest cost is optimal for (21). □

Solving the row-generating PSP becomes increasingly more time consuming as the number of Benders cuts in (**SR-BMP**) grows. An additional complicating factor is that the search for an optimal solution of (21) is restricted to the set of extreme points of the polyhedron described by (21b)- (21d). However, in many applications – including the time-constrained routing problem in Section 4 – (21) is amenable to a simple and very fast solution approach.

We conclude this section by explaining how the existing Benders cuts in (**SR-BMP**) are updated as a result of the incidentally completed minimal variable sets in the $y-$PSP. In this context, the only concern is to preserve the validity of the Benders cuts with respect to $\bar{\Phi}^{s+1}$ following the selection of the new $y-$variables to be included in (**SR-BMP**). Therefore, we only need to identify a feasible solution to (21b)-(21e) for each incidentally completed minimal variable set, $p \in P^t_{\bar{\Phi}^s}$, and $q \in Q^t_{\bar{\Phi}^s}$. To this end, it is sufficient to solve (21) with an empty objective function.

3.4 Convergence Analysis. We next formally discuss that our integrated Benders decomposition and simultaneous column-and-row generation algorithm obtains an optimal solution to (**BMP**). The algorithmic flow depicted in Figure 1 dictates that the algorithm terminates if no $y-$variable prices out favorably in the row-generating PSP. Two conditions are fulfilled when the row-generating PSP is invoked: (i) we have a basic optimal solution to (**SR-BMP**) available, and (ii) the true objective function value of this solution is estimated correctly by the set of Benders cuts in (**SR-BMP**). Based on these conditions, our main convergence theorem rests on three intermediate steps. In the first step, we show

that the basic optimal solution of (**SR-BMP**) on hand is also a basic feasible solution to (**S-BMP**). This step guarantees that without adding new Benders cuts, we obtain an optimal solution to (**S-BMP**) as soon as $\zeta_{yx} \geq 0$ and the reduced costs of all columns become non-negative. The second step proves that after augmenting the optimal solution of (**SR-BMP**) with a family of minimal variable sets, the current optimal solution of (**SR-DBSP**) is also properly augmented; that is, the augmented solution is optimal for the *new* (**SR-DBSP**). Then, we show in the third step that after lifting an extreme point (ray), we still have an extreme point (ray) for the new altered dual polyhedron. With this step, we ensure that the representation theorem is applied correctly within the master problem and guarantee the finite convergence of our algorithm. In the sequel, we provide a series of intermediate results before formally proving the correctness of our algorithm in Theorem 3.1.

If $\zeta_{yx} < 0$, then the row-generating PSP determines that there exists $k \in K \setminus \bar{K}$ such that $\bar{c}_k = \zeta_{yx} < 0$ if $y_l, l \in \Sigma_k = \cup_{S_K^k \in \mathcal{F}_k} S_K^k$ are incorporated into (**SR-BMP**). With this addition, the index sets $\bar{K}$, $\bar{N}$, and $\bar{I}$ are updated as $\bar{K} \cup \Sigma_k$, $\bar{N} \cup S_N(\Sigma_k)$, and $\bar{I} \cup \Delta(S_K^k)$, respectively. The associated changes are reflected in (**SR-BMP**), (**SR-BSP**), and (**SR-DBSP**), and the resulting updated formulations are referred to as **SR-BMP**($\mathbf{\Sigma_k}$), **SR-BSP**($\mathbf{\Sigma_k}$), and **SR-DBSP**($\mathbf{\Sigma_k}$), respectively.

Lemma 3.3 (Primal Basic Feasibility of the Master Problem) *The following two statements hold for* **SR-BMP**($\mathbf{\Sigma_k}$)*: (i) The existing Benders cuts corresponding to* $p \in P^t_{\bar{\Phi}^s}$ *and* $q \in Q^t_{\bar{\Phi}^s}$ *are satisfied by* $\bar{\boldsymbol{y}}' = \begin{pmatrix} \bar{\boldsymbol{y}}^t & \mathbf{0} \end{pmatrix}$*, where the current optimal solution* $\bar{\boldsymbol{y}}^t$ *of* (**SR-BMP**) *is augmented with zeros for* $y_l, l \in \Sigma_k \setminus \bar{K}$*. (ii)* $\bar{\boldsymbol{y}}'$ *is a basic feasible solution for* **SR-BMP**($\mathbf{\Sigma_k}$)*.*

Proof. Recall that the outcome of the row-generating PSP modifies the existing Benders cuts in (**SR-BMP**) as indicated in (18) for an optimality cut. Therefore, to show part (i), it is sufficient to demonstrate that the *new* term in (18) vanishes and the *updated* term reduces to its original state. As the modifications applied to a feasibility cut are identical to those applied to an optimality cut, the proof is valid for both types of cuts. Let us start with the *new* term. Note that for $S_K^k \in \mathcal{F}_k$ and for both $r \in P^t_{\bar{\Phi}^s}$ and $r \in Q^t_{\bar{\Phi}^s}$, we have $w_i^r = 0$, $i \in \Delta_0(S_K^k)$ by constraints (21c) and $r_i = 0$, $i \in \Delta_+(S_K^k)$ by Assumption A.3. Thus, we obtain

$$\sum_{i \in \Delta(\Sigma_k)} r_i w_i^r = \sum_{S_K^k \in \mathcal{F}_k} \sum_{i \in \Delta(S_K^k)} r_i w_i^r = \sum_{S_K^k \in \mathcal{F}_k} \sum_{i \in \Delta_0(S_K^k)} r_i w_i^r + \sum_{S_K^k \in \mathcal{F}_k} \sum_{i \in \Delta_+(S_K^k)} r_i w_i^r = 0.$$

Next, we check the *updated* term. Recall that the formulation of the row-generating PSP enforces $C_{il} w_i^r = 0$ for $l \in \bar{K}$, $i \in \Delta(\Sigma_k)$ for both $r \in P^t_{\bar{\Phi}^s}$ and $r \in Q^t_{\bar{\Phi}^s}$. Furthermore, $\bar{y}'_l = 0$ for $l \in \Sigma_k \setminus \bar{K}$. Therefore,

$$\sum_{l \in \bar{K} \cup \Sigma_k} \sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} C_{il} w_i^r \bar{y}'_l = \sum_{l \in \bar{K}} \sum_{i \in \bar{I}} C_{il} w_i^r \bar{y}'_l + \sum_{l \in \bar{K}} \sum_{\Delta(\Sigma_k)} C_{il} w_i^r \bar{y}'_l + \sum_{l \in \Sigma_k \setminus \bar{K}} \sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} C_{il} w_i^r \bar{y}'_l = \sum_{l \in \bar{K}} \sum_{i \in \bar{I}} C_{il} w_i^r \bar{y}'_l.$$

This completes the proof of part (i).

Since none of the constraint coefficients – except for those in the Benders cuts – are affected by the outcome of the row-generating PSP, $\bar{\boldsymbol{y}}'$ is a feasible solution for **SR-BMP**($\mathbf{\Sigma_k}$) based on part (i). In order to prove part (ii), it then suffices to argue that the coefficient $\sum_{i \in \bar{I}} C_{il} w_i^r$ of a basic variable $l \in \bar{K}$ in a Benders cut $r \in P^t_{\bar{\Phi}^s}$ or $r \in Q^t_{\bar{\Phi}^s}$ remains intact after solving the row-generating PSP. Observe that

$$\sum_{i \in \bar{I} \cup \Delta(\Sigma_k)} C_{il} w_i^r = \sum_{i \in \bar{I}} C_{il} w_i^r + \sum_{i \in \Delta(\Sigma_k)} C_{il} w_i^r = \sum_{i \in \bar{I}} C_{il} w_i^r,$$

because $C_{il}w_i^r = 0$ for $l \in \bar{K}$, $i \in \Delta(\Sigma_k)$ for both $r \in P^t_{\bar{\Phi}^s}$ and $r \in Q^t_{\bar{\Phi}^s}$. □

Lemma 3.3 bears significance from a computational point of view because it implies that **SR-BMP**($\mathbf{\Sigma_k}$) may be warm started from the most recent basic optimal solution of (**SR-BMP**). Moreover, y_k is the only possible candidate for the entering variable in the simplex method as stipulated by the next lemma. In the remainder of this section, $\bar{\boldsymbol{y}}'$ refers to the optimal solution of the current (**SR-BMP**) augmented by zeros for $y_l, l \in \Sigma_k \setminus \bar{K}$ as defined in Lemma 3.3.

Lemma 3.4 (Dual Feasibility of the Master Problem) *The reduced cost of any variable $y_{l'}, l' \in K \setminus \{k\}$ is non-negative with respect to the initial basic feasible solution $\bar{\boldsymbol{y}}'$ of* **SR-BMP**($\mathbf{\Sigma_k}$).

Proof. Since the objective function coefficients of the basic variables and the basis matrix are identical in both the optimal solution of (**SR-BMP**) and the starting solution of **SR-BMP**($\mathbf{\Sigma_k}$), the values of the dual variables $u_j, j \in J$, $\gamma_p, p \in P^t_{\bar{\Phi}^s}$, and $\mu_q, q \in Q^t_{\bar{\Phi}^s}$ remain constant. For any $l' \in K \setminus \{k\}$, the reduced cost of $y_{l'}$ with respect to the initial basic feasible solution of **SR-BMP**($\mathbf{\Sigma_k}$) is then computed as:

$$\begin{aligned}
\bar{c}_{l'} &= c_{l'} - \textstyle\sum_{j\in J} A_{jl'}u_j - \sum_{p\in P^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}\cup\Delta(\Sigma_k)} C_{il'}w_i^p\gamma_p - \sum_{q\in Q^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}\cup\Delta(\Sigma_k)} C_{il'}w_i^q\mu_q \\
&= c_{l'} - \textstyle\sum_{j\in J} A_{jl'}u_j - \sum_{p\in P^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}} C_{il'}w_i^p\gamma_p - \sum_{p\in P^t_{\bar{\Phi}^s}} \sum_{i\in \Delta(\Sigma_k)} C_{il'}w_i^p\gamma_p \\
&\quad - \textstyle\sum_{q\in Q^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}} C_{il'}w_i^q\mu_q - \sum_{q\in Q^t_{\bar{\Phi}^s}} \sum_{i\in \Delta(\Sigma_k)} C_{il'}w_i^q\mu_q \\
&= c_{l'} - \textstyle\sum_{j\in J} A_{jl'}u_j - \sum_{p\in P^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}} C_{il'}w_i^p\gamma_p - \sum_{q\in Q^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}} C_{il'}w_i^q\mu_q,
\end{aligned}$$

which is identical to the reduced cost of $y_{l'}$ in the optimal solution of (**SR-BMP**) and must be non-negative. The last equality follows from the same argument used in the proof of Lemma 3.3 – $C_{il'} = 0$ for $i \in \Delta_+(\Sigma_k)$, $l' \neq k$ and $w_i^p = w_i^q = 0$ for $i \in \Delta_0(\Sigma_k)$, $p \in P^t_{\bar{\Phi}^s}$, $q \in Q^t_{\bar{\Phi}^s}$. □

In the next lemma, we show that an optimal solution for **SR-DBSP**($\mathbf{\Sigma_k}$) can be constructed from the optimal solution of (**SR-DBSP**). Moreover, both have the same objective function value.

Lemma 3.5 (Optimality of the Subproblem) *An optimal solution to* **SR-DBSP**($\mathbf{\Sigma_k}$) *solved for $\bar{\boldsymbol{y}}'$ is given by $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}')$, where $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ is an optimal solution of* (**SR-DBSP**) *solved for $\bar{\boldsymbol{y}}^t$ and the values $\bar{w}'_i$, $i \in \Delta(\Sigma_k)$ are set by solving* (21) *for $S^k_K \in \mathcal{F}_k$ given $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$. In addition, we have $z(\bar{\boldsymbol{y}}') = z(\bar{\boldsymbol{y}}^t)$.*

Proof. Given $\bar{\boldsymbol{y}}^t$, let $\bar{\boldsymbol{x}}^t$ denote the optimal solution of (**SR-BSP**). We then construct an optimal solution to **SR-BSP**($\mathbf{\Sigma_k}$) given $\bar{\boldsymbol{y}}'$. Since $k \notin \bar{K}$ by definition, $\bar{y}'_k = 0$ holds and the new constraints in **SR-BSP**($\mathbf{\Sigma_k}$) indexed by $\Delta_+(\Sigma_k)$ involving y_k force the values of the new variables x_n, $n \in S_N(\Sigma_k)$ to zero by Assumption A.3. We claim that $\bar{\boldsymbol{x}}' = \begin{pmatrix}\bar{\boldsymbol{x}}^t & \mathbf{0}\end{pmatrix}$, where the current optimal solution $\bar{\boldsymbol{x}}^t$ of (**SR-BSP**) is augmented with zeros for $x_n, n \in S_N(\Sigma_k)$, is a feasible solution for **SR-BSP**($\mathbf{\Sigma_k}$). The constraints of **SR-BSP**($\mathbf{\Sigma_k}$) indexed by $m \in M$ and $i \in \bar{I}$ are satisfied because $\bar{x}'_n = 0, n \in S_N(\Sigma_k)$, $\bar{y}'_l = 0, l \in \Sigma_k \setminus \bar{K}$, and $\bar{\boldsymbol{x}}^t$ is feasible for (**SR-BSP**). In addition, the second and third assumptions in Appendix A collectively imply that the new linking constraints in **SR-BSP**($\mathbf{\Sigma_k}$) indexed by $i \in \Delta(\Sigma_k)$ are not violated by $\bar{x}'_n = 0, n \in S_N(\Sigma_k)$ if $\bar{y}'_l = 0, l \in \Sigma_k \setminus \bar{K}$. Thus, we conclude that $\bar{\boldsymbol{x}}'$ is a feasible solution to **SR-BSP**($\mathbf{\Sigma_k}$) with an objective function value of $z(\bar{\boldsymbol{y}}^t)$.

We next demonstrate that $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}')$ is feasible for **SR-DBSP**($\mathbf{\Sigma_k}$) with the same objective function value $z(\bar{\boldsymbol{y}}^t)$. By strong duality, this certifies $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}')$ as optimal for **SR-DBSP**($\mathbf{\Sigma_k}$) and proves

$z(\bar{\boldsymbol{y}}') = z(\bar{\boldsymbol{y}}^t)$. To this end, observe that the variables $w_i, i \in \Delta(\Sigma_k)$ do not appear in the constraints of **SR-DBSP**$(\boldsymbol{\Sigma_k})$ indexed by $\bar{N}$ due to Assumption A.1. Consequently, $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}')$ fulfills these constraints because $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ is a feasible solution of (**SR-DBSP**). The remaining constraints in **SR-DBSP**$(\boldsymbol{\Sigma_k})$ indexed by $S_N(\Sigma_k)$ and the non-negativity constraints for $w_i, i \in \Delta(\Sigma_k)$ are imposed by (21) and hold for $\bar{\boldsymbol{w}}'$. Therefore, the constructed solution $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}, \bar{\boldsymbol{w}}')$ is feasible for **SR-DBSP**$(\boldsymbol{\Sigma_k})$. The associated objective function value is $z(\bar{\boldsymbol{y}}^t)$, because the terms labeled as *updated* and *new* in (18) do not alter the objective function evaluation as demonstrated in the proof of part (i) of Lemma 3.3. □

The following lemma shows that after lifting the existing extreme points and rays of the current dual polyhedron according to the solution of the row-generating PSP, the resulting augmented points and rays are also extremes for the new dual polyhedron. This is a critical step as the representation theorem dictates to work with extreme points and rays, and is a key result for the finite convergence of our algorithm. For ease of presentation, we restrict our attention to **SR-DBSP**$(\mathbf{S^k_K})$ in the Lemma 3.6 below. The result is equally valid for **SR-DBSP**$(\boldsymbol{\Sigma_k})$ by applying the construction described below iteratively for each $S^k_K \in \mathcal{F}_k$. For the purposes of Lemma 3.6, $\mathbf{F}$ represents the coefficients of the variables $v_m, m \in M$ and $w_i, i \in \bar{I}$ in the linearly independent constraints of (**SR-DBSP**) active at some extreme point $(\bar{\boldsymbol{v}}^p, \bar{\boldsymbol{w}}^p) \in P^t_{\bar{\Phi}^s}$. The matrices $\mathbf{G}$ and $\mathbf{H}$ are formed by the coefficients associated with the new constraints in **SR-DBSP**$(\mathbf{S^k_K})$ active at $(\bar{\boldsymbol{v}}^p, \bar{\boldsymbol{w}}^p, \bar{\boldsymbol{w}}^{p\prime})$ – designated by p' in the following discussion, where $\bar{w}^{p\prime}_i$, $i \in \Delta(S^k_K)$ are determined by solving (21) for S^k_K and p. The following matrix shows the structure of these coefficients in **SR-DBSP**$(\mathbf{S^k_K})$:

$$\left(\begin{array}{c|c} \mathbf{F} & \mathbf{0} \\ \hline \mathbf{G} & \mathbf{H} \end{array}\right), \tag{22}$$

where $\mathbf{H}$ denotes the coefficients of $w_i, i \in \Delta(S^k_K)$ in the new constraints in **SR-DBSP**$(\mathbf{S^k_K})$. Observe that these variables do not appear in the constraints indexed by $n \in \bar{N}$ based on Assumption A.1. Finally, note that we obtain a very similar construction for an extreme ray $(\bar{\boldsymbol{v}}^q, \bar{\boldsymbol{w}}^q) \in Q^t_{\bar{\Phi}^s}$ and $(\bar{\boldsymbol{v}}^q, \bar{\boldsymbol{w}}^q, \bar{\boldsymbol{w}}^{q\prime})$ – denoted by q' in the sequel, where solving (21) for S^k_K and q yields $\bar{w}^{q\prime}_i$, $i \in \Delta(S^k_K)$. The only difference is that the matrices $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are now defined with respect to the constraints tight at q and q' in the recession cones of $\bar{\Phi}^s$ and $\bar{\Phi}^{s+1}$, respectively. Consequently, $\mathbf{F}$ is a $(\delta - 1) \times \delta$ matrix of rank $\delta - 1$ instead of a $\delta \times \delta$ square matrix of full rank, where $\delta = \mid M \mid + \mid \bar{I} \mid$ is the dimension of $\bar{\Phi}^s$.

Lemma 3.6 (Validity of Lifting) *Let p and q be an extreme point and an extreme ray of the dual polyhedron $\bar{\Phi}^s$ associated with* (**SR-DBSP**)*, respectively. Suppose for a given S^k_K, we construct p' and q' out of p and q, respectively, according to the solution of* (21)*. Then, p' is an extreme point and q' is an extreme ray of the dual polyhedron $\bar{\Phi}^{s+1}$ induced by* **SR-DBSP**$(\mathbf{S^k_K})$.

Proof. The point p' is feasible with respect to the constraints indexed by $n \in \bar{N}$ in **SR-DBSP**$(\mathbf{S^k_K})$ because the variables $w_i, i \in \Delta(S^k_K)$ do not factor into these, and p is feasible for (**SR-DBSP**). The constraints (21b)-(21d) ensure that p' is feasible with respect to the remaining constraints of **SR-DBSP**$(\mathbf{S^k_K})$. The δ linearly independent constraints active at p denoted by the $\delta \times \delta$ square matrix $\mathbf{F}$ in (22) are also binding at p'. Furthermore, the constraint (21e) assures that the rows of the $\mid \Delta(S^k_K) \mid \times \mid \Delta(S^k_K) \mid$ square matrix $\mathbf{H}$ are linearly independent. Consequently, p' satisfies $\mid \Delta(S^k_K) \mid$-many additional linearly independent constraints of **SR-DBSP**$(\mathbf{S^k_K})$ – depicted as $\begin{pmatrix} \mathbf{G} & \mathbf{H} \end{pmatrix}$ in (22) – as equality. The dual polyhedron $\bar{\Phi}^{s+1}$ is of dimension $\delta + \mid \Delta(S^k_K) \mid$, and in order to conclude that p' is an extreme point of $\bar{\Phi}^{s+1}$

it is then sufficient to argue that the $(\delta+ \mid \Delta(S_K^k) \mid) \times (\delta+ \mid \Delta(S_K^k) \mid)$ square matrix given in (22) is of full rank. This is obviously true because both square matrices $\mathbf{F}$ and $\mathbf{H}$ are of full rank.

The same line of reasoning presented above applies analogously to q and q', where the only difference is that the arguments are developed with respect to the recession cones of $\bar{\Phi}^s$ and $\bar{\Phi}^{s+1}$. □

We are now ready to prove the optimality of our proposed approach given in Figure 1.

THEOREM 3.1 (CORRECTNESS AND FINITE CONVERGENCE) *Suppose that at iteration t, we have the sets $\bar{K}$, $\bar{N}$, and $\bar{I}$. Let the pair $(\bar{\boldsymbol{y}}^t, \bar{z}^t)$ be the optimal solution of* (**SR-BMP**)*. If the row-generating PSP yields the optimal objective function value $\zeta_{yx} \geq 0$, then $(\bar{\boldsymbol{y}}', \bar{z}^t)$ is optimal for* (**BMP**)*, where $\bar{\boldsymbol{y}}' = \begin{pmatrix} \bar{\boldsymbol{y}}^t & \mathbf{0} \end{pmatrix}$ is obtained by augmenting $\bar{\boldsymbol{y}}^t$ with zeros for $y_k, k \in K \setminus \bar{K}$.*

PROOF. Observe that none of the Lemmas 3.3–3.6 requires an optimal solution to (21). All results presented so far in this section are valid as long as we pick some extreme point of the polyhedron specified by (21b)-(21d) in each call to (21) while solving the row-generating PSP. The particular choice of the extreme point does only impact the reduced cost of the variable $k \notin \bar{K}$ currently under consideration.

Based on the observation in the previous paragraph, we can construct a basic feasible solution of (**S-BMP**) out of the basic optimal solution of (**SR-BMP**) at iteration t. To this end, select an arbitrary family $\mathcal{F}$ of index sets S_K such that $(K \setminus \bar{K}) \subseteq \Sigma = \cup_{S_K \in \mathcal{F}} S_K$. Different from the row-generating PSP, these minimal variable sets do not need to share a common variable index, because we are not interested in minimizing the reduced cost of a particular variable at this step. For each $S_K \in \mathcal{F}$, solve (21) with an empty objective function for each $p \in P^t_{\bar{\Phi}^s}$ and $q \in Q^t_{\bar{\Phi}^s}$, where $\bar{\Phi}^s$ represents the dual polyhedron at iteration t. Then, by applying Lemma 3.3 to Σ, we conclude that $\bar{\boldsymbol{y}}'$ is a basic feasible solution to (**S-BMP**).

Next, we argue that no variable $y_k, k \in K \setminus \bar{K}$ prices out favorably in (**S-BMP**) with respect to the basis associated with $\bar{\boldsymbol{y}}'$ if $\zeta_{yx} \geq 0$. To this end, note that all minimal variable sets of the form $\{y_l \mid l \in S_K^k\}$ absent from (**SR-BMP**) in iteration t are included in (**S-BMP**). The corresponding family of index sets is clearly an element of $\mathcal{P}_k$. Thus, we conclude that the reduced cost $\bar{c}_k$ of y_k in (**S-BMP**) is non-negative because $0 \leq \zeta_{yx} \leq \zeta^k_{yx} \leq \bar{c}_k$, where ζ^k_{yx} denotes the minimum reduced cost of y_k computed over all possible members of $\mathcal{P}_k$ in the row-generating PSP. Hence, $\bar{\boldsymbol{y}}'$ is a basic optimal solution of (**S-BMP**).

Each minimal variable set in $\mathcal{F}$ features at least one variable currently missing from (**SR-BMP**). Therefore, Lemma 3.5 can be invoked iteratively for all $S_K \in \mathcal{F}$. This process keeps augmenting $\bar{N}$ and $\bar{I}$ until all missing constraints indexed by $N \setminus \bar{N}$ and all missing variables $w_i, i \in I(K, N) \setminus \bar{I}$ are incorporated back into the dual polyhedron – ultimately yielding (**DBSP**) solved over Φ. Furthermore, Lemma 3.5 guarantees that the optimal objective function value of (**DBSP**) solved for $\bar{\boldsymbol{y}}'$ is equal to $z(\bar{\boldsymbol{y}}^t)$ so that $z(\bar{\boldsymbol{y}}') = z(\bar{\boldsymbol{y}}^t) \leq \bar{z}^t$ holds based on the flow in Figure 1.

Finally, an iterative application of Lemma 3.6 to each $S_K \in \mathcal{F}$ assures us that the extreme points and extreme rays of $\bar{\Phi}^s$ associated with (**SR-DBSP**) are correctly lifted to become extreme points and extreme rays of Φ associated with (**DBSP**). Thus, all Benders feasibility and optimality cuts in (**S-BMP**) are valid with respect to the original dual polyhedron Φ and their number is finite. Moreover, the relation

$z(\bar{\boldsymbol{y}}') \leq \bar{z}^t$ implies that no Benders cut $p \in P_\Phi$ or $q \in Q_\Phi$ is violated by the optimal solution of (**S-BMP**), and the algorithm terminates with an optimal solution $(\bar{\boldsymbol{y}}', \bar{z}^t)$ to (**BMP**) in a finite number of iterations. □

4. Time-Constrained Routing Problem. In this section, the mechanics and promise of our approach is exemplified through an application to a time-constrained routing (TCR) problem. We first define this problem and then solve its LP relaxation through our integrated Benders decomposition and simultaneous column-and-row generation algorithm. The computational results attest to the effectiveness of the proposed solution framework.

An application that needs to schedule the visit of a tourist to a given geographical area as efficiently as possible motivates the time-constrained routing (TCR) problem defined and studied by Avella et al. (2006). The tourist must be sent on one tour on each day of the vacation period so that his/her total satisfaction is maximized and no attraction is visited more than once.

TCR can be formulated as a set packing problem with side constraints. To be consistent with (**MP**), we change the notation given in Avella et al. (2006). The set of sites that may be visited by a tourist in a vacation period M is denoted by J, and K represents the set of daily tours that originate from and terminate at the same location. Each tour is a sequence of sites to be visited on the same day, provided that it satisfies the time-windows restrictions of the tourist and the other feasibility criteria. The total satisfaction of the tourist from participating in tour k is given by c_k, and the binary variable y_k is set to one, if tour k is incorporated into the itinerary of the tourist. If tour k is performed on day m, then the binary variable x_{km} takes the value one. The overall mathematical model is given as

$$\text{maximize} \quad \sum_{k \in K} c_k y_k, \tag{23}$$

$$\text{subject to} \quad \sum_{k \in K} A_{jk} y_k \leq 1, \qquad j \in J, \tag{24}$$

$$\sum_{k \in K} B_{km} x_{km} \leq 1, \qquad m \in M, \tag{25}$$

$$y_k - \sum_{m \in M} D_{km} x_{km} = 0, \qquad k \in K, \tag{26}$$

$$\sum_{k \in K} y_k = |M|, \tag{27}$$

$$y_k \in \{0, 1\}, \qquad k \in K, \tag{28}$$

$$x_{km} \in \{0, 1\}, \qquad k \in K, m \in M. \tag{29}$$

The matrix $\mathbf{A}$ is binary, and $A_{jk} = 1$ only if tour k contains site j. The matrices $\mathbf{B}$ and $\mathbf{D}$ are identical, and $B_{km} = D_{km} = 1$ indicates that tour k can be performed on day m; $B_{km} = D_{km} = 0$ otherwise. Due to the constraints (24), at most one tour in the selected itinerary of the tourist visits site j. The linking constraints (26) impose that a tour to be included in the itinerary is assigned to one of the days allowed in M, and we also require that exactly one tour is selected on each day of the vacation period as prescribed jointly through the constraints (25)-(27). The constraints (25), which were originally presented as equality constraints in Avella et al. (2006), are posed here as inequality constraints without affecting the validity of the formulation and help us circumvent the technical difficulties arising from the existence of a line in the dual polyhedron. Finally, the objective (23) maximizes the total satisfaction of the tourist

over the vacation period M.

Avella et al. (2006) solve the LP relaxation of (23)-(29) by a column-and-row generation approach due to a potentially huge number of tours. Muter et al. (2012) demonstrate that the optimality condition of Avella et al. (2006) for their column-and-row generation approach leads to a premature termination of the algorithm and prove a correct stopping condition.

4.1 Application of the Proposed Methodology. In our integrated Benders decomposition and simultaneous column-and-row generation framework, the LP relaxation of TCR is re-formulated as

$$\begin{array}{lll} \text{maximize} & \sum_{k\in K} c_k y_k + z(\boldsymbol{y}), & \\ \text{subject to} & \sum_{k\in K} A_{jk} y_k \leq 1, & j \in J, \\ & \sum_{k\in K} y_k = |M|, & \\ & y_k \geq 0, & k \in K, \end{array} \tag{30}$$

where the Benders subproblem becomes

$$\begin{array}{llll} z(\bar{\boldsymbol{y}}) = & \text{maximize} & 0, & \\ & \text{subject to} & \sum_{k\in K} B_{km} x_{km} \leq 1, & m \in M, \\ & & \sum_{m\in M} D_{km} x_{km} = \bar{y}_k, & k \in K, \\ & & x_{km} \geq 0, & k \in K, m \in M. \end{array} \tag{31}$$

Clearly, this is only a feasibility problem, and we only check whether there exists a feasible solution to (31) given the values of the $y-$variables. The dual slave problem is then formulated as

$$\begin{array}{llll} z(\bar{\boldsymbol{y}}) = & \text{minimize} & \sum_{k\in K} \bar{y}_k w_k + \sum_{m\in M} v_m, & \\ & \text{subject to} & w_k + v_m \geq 0, & k \in K, m \in M^k, \\ & & v_m \geq 0, & m \in M, \end{array} \tag{32}$$

where $\boldsymbol{v}$ and $\boldsymbol{w}$ are the sets of dual variables corresponding to the first and the second set of constraints in (31) respectively, and $M^k = \{m \in M \mid B_{km} = 1\}$. The dual polyhedron Φ is always non-empty because $\boldsymbol{v} = \boldsymbol{w} = \boldsymbol{0}$ is feasible regardless of $\bar{\boldsymbol{y}}$. Furthermore, Φ is a polyhedral cone with a unique extreme point at the origin. Therefore, (**BMP**) is formed only with the feasibility cuts:

$$\begin{array}{lll} \text{maximize} & \sum_{k\in K} c_k y_k, & \\ \text{subject to} & \sum_{k\in K} A_{jk} y_k \leq 1, & j \in J, \\ & \sum_{k\in K} y_k = |M|, & \\ & \sum_{k\in K} w_k y_k \geq -\sum_{m\in M} v_m, & (\boldsymbol{v}, \boldsymbol{w}) \in Q_\Phi, \\ & y_k \geq 0, & k \in K. \end{array} \tag{33}$$

Mercier (2008) performs a theoretical comparison of the different types of feasibility cuts for the integrated aircraft-routing and crew-pairing problem. The Benders subproblem in this work is a feasibility problem; however, the author argues that the identification of a ray of unboundedness "may be a difficult task, especially if the primal form of the subproblem is being solved." Therefore, Mercier adds artificial variables to the Benders subproblem and guarantees its feasibility and the boundedness of the dual slave

problem. Feasibility cuts associated with the original Benders subproblem are then obtained from the extreme point optimal solutions of the modified Benders subproblem. Applying this strategy to (31) leads to the following modified Benders subproblem

$$\begin{aligned} z(\bar{\boldsymbol{y}}) = \quad & \text{maximize} && \sum_{m\in M} -s_m^+, \\ & \text{subject to} && \sum_{k\in K} B_{km}x_{km} - s_m^+ \le 1, && m \in M, \\ & && \sum_{m\in M} D_{km}x_{km} = \bar{y}_k, && k \in K, \\ & && x_{km} \ge 0, && k \in K, m \in M, \\ & && s_m^+ \ge 0, && m \in M, \end{aligned} \tag{34}$$

and the modified dual slave problem

$$\begin{aligned} z(\bar{\boldsymbol{y}}) = \quad & \text{minimize} && \sum_{k\in K} \bar{y}_k w_k + \sum_{m\in M} v_m, \\ & \text{subject to} && w_k + v_m \ge 0, && k \in K, m \in M^k, \\ & && 0 \le v_m \le 1, && m \in M. \end{aligned} \tag{35}$$

The Benders master problem imposes that a total of $\mid M \mid$ tours are selected. However, it is oblivious to the requirement that each tour must be assigned to a distinct day $m \in M$ in a feasible solution of the Benders subproblem (31). If such an assignment is not possible for the given set of tours $\bar{\boldsymbol{y}}$ from the master problem, s_m^+ must assume a positive value for some $m \in M$ in the modified Benders subproblem (34) and $z(\bar{\boldsymbol{y}}) < 0$. In such cases, the short Benders master problem is augmented with a violated feasibility cut created from an extreme point optimal solution of (35), where the feasible region of (35) is denoted by Φ_c and its set of extreme points is represented by P_{Φ_c}. The following proposition characterizes the relationship between Q_Φ and P_{Φ_c} and justifies solving the modified dual slave problem (35) instead of the original dual slave problem (32).

Proposition 4.1 *Each extreme ray of unboundedness in Q_Φ corresponds to an extreme point in P_{Φ_c}.*

Proof. See Appendix B. □

Observe that (35) is always feasible and bounded for any $\bar{\boldsymbol{y}}$, and any extreme point optimal solution $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}) \in P_{\Phi_c}$ with $z(\bar{\boldsymbol{y}}) < 0$ gives rise to a feasibility cut $\sum_{k\in K} \bar{w}_k y_k \ge -\sum_{m\in M} \bar{v}_m$ in the short Benders master problem. The convergence of the algorithm is guaranteed because $Q_\Phi \subseteq P_{\Phi_c}$ and any extreme point optimal solution $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ of (35) such that $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}) \in P_{\Phi_c} \setminus Q_\Phi$ is still feasible with respect to (32). Thus, such an optimal solution of (35) is still a ray of Φ – although it is not extreme – and leads to a valid feasibility cut.

So far, we have assumed that all tours $k \in K$ are included in the formulation a priori in order to develop the Benders master and (modified) subproblems for TCR. However, the cardinality of K is generally large and motivates us to solve the Benders master problem via our integrated Benders decomposition and simultaneous column-and-row generation framework. To this end, we set up the short-and-restricted Benders master problem with a strict subset $\bar{K}$ of the set of all possible tours. Observe that each $y-$variable appears in a unique linking constraint of the form (26). Therefore, the short-and-restricted modified dual slave problem is initialized only with the variables $w_k, k \in \bar{K}$ associated with the

existing linking constraints and the constraints associated with $x_{km}, k \in \bar{K}, m \in M^k$. As usual, the dual polyhedron is updated after the solution of the row-generating PSP developed in the sequel and labeled as $\bar{\Phi}_c^s$ after s updates.

Each variable $y_k, k \in K$ in TCR appears in a distinct linking constraint and forms a unique minimal variable set $S_K^k = \{k\}$. In other words, $\mathcal{F}_k = \{\{k\}\}$ and $\Sigma_k = \{k\}$ for all $k \in K \setminus \bar{K}$. The expression (20) for the reduced cost of a $y-$variable then reduces to

$$\begin{aligned}\bar{c}_k =& c_k - \sum_{j\in J} A_{jk}u_j - \sum_{p\in P^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}\cup\Delta(\Sigma_k)} C_{ik}w_i^p\gamma_p - \sum_{q\in Q^t_{\bar{\Phi}^s}} \sum_{i\in \bar{I}\cup\Delta(\Sigma_k)} C_{ik}w_i^q\mu_q \\ =& c_k - \sum_{j\in J} A_{jk}u_j - \sum_{p\in P^t_{\bar{\Phi}^s_c}} C_{kk}w_k^p\gamma_p \end{aligned} \tag{36}$$

for $k \notin \bar{K}$, where the kth linking constraint is associated with y_k and $C_{kk} = 1$ is the coefficient of y_k in this currently missing constraint. In contrast with (16), all terms associated with the existing linking constraints drop out, and the $y-$PSP becomes redundant in the context of TCR. Based on these observations, the row-generating PSP is formulated as

$$\zeta_{yx} = \max_{k\in(K\setminus\bar{K})} \left\{ c_k - \sum_{j\in J} A_{jk}u_j - \theta - \sum_{p\in P^t_{\bar{\Phi}^s_c}} \gamma_p\alpha_k(p) \right\},$$

where

$$\begin{aligned} \alpha_k(p) =& \text{minimize} && w_k^p, && \text{(37a)} \\ & \text{subject to} && w_k^p \geq -v_m^p, \quad m \in M^k, && \text{(37b)} \\ & && \text{One tight constraint among (37b).} && \text{(37c)} \end{aligned}$$

Note that we must also take into account the dual variable θ corresponding to the cardinality constraint defined over the $y-$variables in (**SR-BMP**). For each $p \in P^t_{\bar{\Phi}^s_c}$, the optimal solution of (37a)-(37c) is $w_k^p = \max_{m\in M^k}\{-v_m^p\} = -\min_{m\in M^k}\{v_m^p\}$. Thus, the row-generating PSP for TCR simplifies to

$$\zeta_{yx} = \max_{k\in K\setminus\bar{K}} \left\{ c_k - \sum_{j\in J} A_{jk}u_j - \theta + \sum_{p\in P^t_{\bar{\Phi}^s_c}} \gamma_p \min_{m\in M^k}\{v_m^p\} \right\}.$$

If $\zeta_{yx} > 0$, the variable y_k with the maximum reduced cost is added to (**SR-BMP**), and the Benders cuts in (**SR-BMP**) are updated by adding the term $C_{kk}w_k^p y_k = -\min_{m\in M^k}\{v_m^p\}y_k$ for each $p \in P^t_{\bar{\Phi}^s_c}$. Otherwise, the optimal solution for (**BMP**) is attained as prescribed in Figure 1.

We conclude this section by discussing the choices regarding the $x-$PSP. It turns out that the $x-$PSP is optional from a practical point of view because the number of days in a vacation period is typically small. Thus, for any $y_k, k \notin \bar{K}$ to be incorporated as a result of solving the row-generating PSP we can include all constraints for $x_{km}, m \in M^k$ into (**SR-DBSP**) along with w_k. However, it is evident from the structure of the optimal solution of the row-generating PSP that some of these constraints would be redundant. Thus, we can also opt for only adding the tight constraints among (37b) and then invoking the $x-$PSP by following the flow in Figure 1 in order to price out the variables x_{km} currently excluded from (**SR-BSP**), where $k \in \bar{K}$. In the computational experiments in the next section, we adopt the first approach.

4.2 Computational Experiments. In this section, we present the results of our computational experiments on a set of randomly generated TCR instances. The experiments are conducted on a personal computer with a 3.60 GHz Intel Xeon E5-1620 processor and 16 GB of RAM. Our codes are implemented in Visual C++ by employing `IBM ILOG CPLEX 12.5` with the default settings and the `Concert Technology` component library to solve the LP relaxation of TCR in three different ways. We compare the performance of the simultaneous column-and-row generation algorithm developed in Muter et al. (2013a) (*CRG*) with that of the proposed algorithm in this paper in order to justify our efforts in integrating simultaneous column-and-row generation with Benders decomposition (*CRG-BDS*). The complete set of instances are also solved directly by `CPLEX` to set a benchmark. We show that *CRG-BDS* is the best-performing algorithm on average, and `CPLEX` is outperformed by the CRG-based algorithms.

The algorithmic flow illustrated in Figure 1 defers solving the row-generating PSP until no more Benders cuts are generated and reflects the general rationale that the row-generating PSP is inherently difficult to solve. However, in our particular implementation for TCR, the row generating PSP boils down to a linear-time search. Therefore, the performance of our algorithm is improved by deviating from the general flow in Figure 1 as follows: we keep solving the row-generating PSP until no new y-variable is generated and then proceed with the Benders subproblem. The algorithm stops if the objective function value of the Benders subproblem is zero; that is, if a feasible daily assignment is determined for the tours selected in the Benders master problem. Otherwise, the row-generating PSP is re-invoked. Furthermore, $\bar{K}$ is empty at the start of the algorithm, and the feasibility of (**SR-BMP**) is initially guaranteed by a single non-negative artificial variable inserted into the left hand side of the second constraint in (33) and penalized with a large negative coefficient in the objective. To demonstrate the typical behavior of *CRG-BDS* as new columns and Benders cuts are incorporated, we plot the objective function value of (**SR-BMP**) over the iterations of the algorithm for a single instance in Figure 2. We observe that the objective function increases monotonically and only the row-generating PSP is solved repeatedly until the artificial variable is driven out of the solution. Following this milestone, the Benders subproblem is called whenever the row-generating PSP fails to identify a negatively-priced $y-$variable, and the objective generally drops as a result of adding a new cut to (**SR-BMP**). Subsequently, the row-generating PSP adds new columns, and the objective value of (**SR-BMP**) is restored to its previous levels.

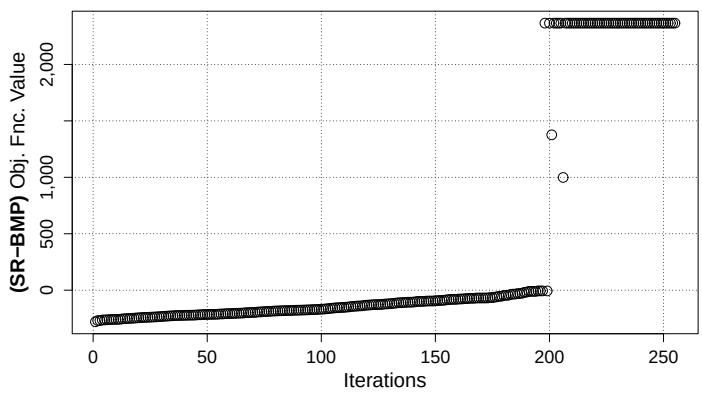

Figure 2: The progress of the objective function values of (**SR-BMP**) over the iterations.

In Avella et al. (2006), the set of possible tours is enumerated by checking several feasibility rules, and some dominance rules are implemented to speed up the tour enumeration. In this paper, our sole concern is contrasting the solution time performances of the algorithms, and the instances are generated differently. The values for the number of sites that can be visited are $|J| \in \{50, 150\}$, and four values are selected for the length of the vacation period so that $|M| \in \{7, 14, 21, 28\}$. The number of sites each tour can visit is generated from an integer uniform distribution $U[1, 10]$, and the sites are picked randomly. Each site is assigned a rating from an integer uniform distribution $U[1, 100]$, and the objective function coefficient of a tour is defined as the sum of the ratings of the sites visited on the tour. Our simple modification to the flow in Figure 1 guarantees correct termination if (**SR-BSP**) can identify a feasible daily assignment of the tours, given the values of the associated $y-$variables retrieved from the solution of (**SR-BMP**). Therefore, the number of days in the vacation period on which tours may take place has an impact on the performance of the proposed methodology. We generate this parameter using an integer uniform distribution $U[1, P]$, and a given tour is then available on a randomly selected set of days of required cardinality in the vacation period. In our preliminary experiments, we set $P \in \{1, 3, 7\}$ and observe that larger values of P benefit our approach and degrade the performance of `CPLEX`. On the one hand, increasing P simplifies the task of determining a feasible daily assignment for the tours in (**SR-BSP**) and expedites the convergence of *CRG-BDS*. On the other hand, higher values of P result in more $x-$variables in the formulation and deteriorates the performance of `CPLEX` in solving the full-blown model. A further insight provided by our preliminary experiments is that the solution time of *CRG* is relatively insensitive to increases in the value of P. In this algorithm, the pricing PSP is the main determinant of the solution time, and the number of columns added through the calls to the pricing PSP does not change drastically with P. These observations collectively lead us to set $P = 3$ for generating the instances used in the comparative analysis with the goal of demonstrating the average performance of the algorithms over relatively tough instances for *CRG-BDS*. The number of tours $|K|$ ranges from 10,000 to 100,000 in increments of 10,000 in the instances created. Five instances are solved for each combination of $|J|$, $|M|$ and $|K|$.

In Tables 2 and 3, the statistics related to *CRG-BDS*, *CRG*, and `CPLEX` are depicted for $|M| \in \{7, 14\}$ and $|M| \in \{21, 28\}$, respectively. For all methods, the total solution time in seconds is indicated under *Soln T*. The total number of the $y-$variables generated by *CRG-BDS* and *CRG* is given in columns labeled as *# Cols*, and for *CRG-BDS* the number of the Benders cuts added to (**SR-BMP**) and the total time expended for solving the row-generating PSP over the course of the algorithm are reported under *# Cuts* and *Pricing T*, respectively. The total time spent by (**SR-BSP**) for creating Benders cuts is negligible and is therefore not presented separately. Thus, the time taken up by (**SR-BMP**) corresponds to the difference between *Soln T* and *Pricing T*. All reported figures are averaged over the five instances created for the corresponding parameter combination. When compared against `CPLEX`, the solution times for *CRG-BDS* and *CRG* are less sensitive to the number of tours and sites. For *CRG-BDS*, even the impact of changing $|J|$ from 50 to 150 is relatively minor for fixed $|M|$, except when $|M| = 14$. The underlying reason is that the optimal solutions for different values of $|K|$ contain roughly the same number of non-zero $y-$ and $x-$variables as long as $|M|$ is fixed. In contrast, the solution times of `CPLEX` increase rapidly with $|K|$. For instance, the average solution times of `CPLEX` and *CRG-BDS* for $|K| = 100,000$ over all instances in Tables 2-3 are 15.6 and 5.6 seconds, respectively. Increasing $|M|$

has an adverse affect on the performances of *CRG* and *CRG-BDS* because the total number of non-zero $y-$ and $x-$variables in the optimal solution increases and more iterations have to be performed. Overall, *CRG-BDS* outperforms the other two algorithms and exhibits the best scaling performance. The average solution times over the entire set of instances are 3.3, 7.5, and 5.0 seconds for *CRG-BDS*, `CPLEX`, and *CRG*, respectively. These figures imply a 127% and 52% speedup over `CPLEX` and *CRG* in favor of *CRG-BDS*, respectively. From a different perspective, by taking the ratios of the corresponding numbers in Tables 2 and 3, we conclude that *CRG* is on average 50% slower than *CRG-BDS*, and the solution times of `CPLEX` are on average twice as large as those of *CRG-BDS*.

Table 2: Results of *CRG-BDS*, CPLEX and *CRG* for $|M| = 7$ and $|M| = 14$.

			CRG-BDS				CPLEX	***CRG***	
|K|	**|J|**	**|M|**	**# Cols**	**# Cuts**	**Pricing T**	**Soln T**	**Soln T**	**Soln T**	**# Cols**
10000	50	7	68.8	2.6	0.1	0.4	0.6	0.7	65.0
20000	50	7	69.6	0.8	0.2	0.5	1.5	1.0	69.0
30000	50	7	72.8	0.6	0.4	0.6	2.2	1.3	69.0
40000	50	7	71.6	1.2	0.5	0.8	3.2	1.6	72.2
50000	50	7	73.2	0.6	0.7	1.0	5.3	1.9	72.4
60000	50	7	72.6	0.2	0.8	1.1	7.3	2.1	71.0
70000	50	7	74.6	1.6	1.0	1.3	9.9	2.5	73.6
80000	50	7	72.6	1.2	1.0	1.3	11.4	2.6	73.2
90000	50	7	71.4	0.4	1.1	1.4	13.3	2.9	73.6
100000	50	7	75.6	1.2	1.4	1.7	14.6	3.3	75.8
		Avg	72.3	1.0	0.7	1.0	6.9	2.0	71.5
10000	150	7	51.6	1.6	0.1	0.3	0.7	0.5	52.6
20000	150	7	58.8	1.8	0.2	0.4	1.5	0.9	60.2
30000	150	7	64.4	1.0	0.3	0.6	2.4	1.3	65.0
40000	150	7	66.4	1.2	0.5	0.8	5.4	1.6	67.6
50000	150	7	71.2	0.4	0.7	1.0	7.5	2.0	75.0
60000	150	7	72.2	0.6	0.7	1.2	9.1	2.1	71.4
70000	150	7	73.2	0.4	1.1	1.4	11.8	2.5	74.0
80000	150	7	71.4	0.2	1.3	1.6	12.5	2.8	73.2
90000	150	7	78.4	0.4	1.5	1.9	15.1	3.3	81.2
100000	150	7	76.8	0.6	1.6	2.1	16.4	3.6	76.8
		Avg	68.4	0.8	0.8	1.1	8.2	2.1	69.7
10000	50	14	121.4	9.2	0.2	0.7	0.6	1.0	117.4
20000	50	14	128.4	5.6	0.5	1.0	1.6	1.5	122.2
30000	50	14	134.2	6.6	0.9	1.4	2.4	2.4	131.2
40000	50	14	134.4	6.8	1.1	1.7	3.6	2.8	131.4
50000	50	14	137.8	8.4	1.4	2.1	5.4	3.3	133.0
60000	50	14	142.4	7.2	1.5	2.3	7.4	3.9	139.4
70000	50	14	142.2	6.8	1.6	2.7	8.6	4.4	140.2
80000	50	14	144.6	3.4	2.2	2.8	10.7	4.9	140.0
90000	50	14	144.6	7.6	2.7	3.4	12.1	5.6	140.4
100000	50	14	144.0	9.0	3.1	3.8	14.2	6.1	140.0
		Avg	137.4	7.1	1.5	2.2	6.7	3.6	133.5
10000	150	14	248.4	2.2	0.3	1.5	0.6	2.4	245.0
20000	150	14	236.8	0.6	0.6	1.7	1.4	3.4	237.0
30000	150	14	230.2	1.2	1.0	2.2	2.8	4.5	227.2
40000	150	14	233.4	0.6	1.4	2.6	4.3	5.1	225.4
50000	150	14	227.0	0.8	1.7	3.2	5.5	6.0	221.2
60000	150	14	224.0	1.2	2.4	3.8	6.3	7.0	227.0
70000	150	14	222.8	0.4	3.2	4.3	8.8	8.0	225.6
80000	150	14	223.2	0.8	3.6	4.8	10.3	8.4	221.2
90000	150	14	223.8	1.4	4.3	5.4	12.0	9.7	227.2
100000	150	14	221.8	0.0	4.5	5.6	13.4	10.0	215.2
		Avg	229.1	0.9	2.3	3.5	6.5	6.5	227.2
		Overall Avg	126.8	2.5	1.3	2.0	7.1	3.5	125.5

Table 3: Results of *CRG-BDS*, CPLEX and *CRG* for $|M| = 21$ and $|M| = 28$.

			CRG-BDS				CPLEX	***CRG***	
|K|	**|J|**	**|M|**	**# Cols**	**# Cuts**	**Pricing T**	**Soln T**	**Soln T**	**Soln T**	**# Cols**
10000	50	21	166.8	19.4	0.4	1.0	0.6	1.2	160.8
20000	50	21	177.6	21.8	1.0	1.7	1.6	1.9	162.6
30000	50	21	177.4	14.2	1.3	2.0	2.5	2.9	174.6
40000	50	21	180.0	27.6	2.7	3.5	3.7	3.9	181.4
50000	50	21	186.6	15.0	2.3	3.2	5.6	4.6	181.0
60000	50	21	188.0	18.6	3.1	4.1	7.3	5.2	179.6
70000	50	21	191.4	13.6	2.8	4.0	8.9	5.8	180.8
80000	50	21	191.0	16.2	3.9	4.7	10.6	6.8	185.8
90000	50	21	196.6	17.0	4.6	5.4	12.0	7.4	188.6
100000	50	21	190.0	23.6	6.2	7.2	14.7	8.4	191.4
		Avg	184.5	18.7	2.8	3.7	6.8	4.8	178.7
10000	150	21	202.6	3.6	0.2	1.3	1.2	1.9	196.2
20000	150	21	213.0	7.2	0.5	1.7	2.6	3.0	208.0
30000	150	21	219.4	5.4	0.9	2.2	4.3	4.4	215.2
40000	150	21	224.0	6.8	1.6	2.8	5.9	5.1	214.6
50000	150	21	223.2	2.0	1.7	3.2	7.9	6.1	215.8
60000	150	21	224.0	7.6	2.6	4.0	10.7	6.9	222.2
70000	150	21	232.6	5.0	3.5	4.7	11.3	8.1	227.4
80000	150	21	227.6	8.6	4.1	5.4	13.5	8.8	224.6
90000	150	21	229.4	5.0	4.3	5.7	19.1	9.8	227.6
100000	150	21	232.6	7.8	5.4	6.6	18.0	10.7	227.4
		Avg	222.8	5.9	2.5	3.8	9.5	6.5	217.9
10000	50	28	194.4	19.8	0.4	1.2	0.6	1.5	187.0
20000	50	28	203.2	39.0	1.9	3.0	1.5	2.6	207.0
30000	50	28	211.8	24.4	2.0	3.1	2.5	3.6	207.4
40000	50	28	219.0	32.4	3.7	4.7	3.6	4.5	214.0
50000	50	28	212.6	31.4	4.4	5.5	5.4	5.3	214.8
60000	50	28	219.6	35.6	5.6	6.9	6.9	6.1	217.4
70000	50	28	233.2	22.8	4.3	5.9	8.1	7.2	219.4
80000	50	28	230.4	27.8	6.3	7.4	10.0	8.0	223.2
90000	50	28	225.8	33.6	8.5	9.6	11.1	8.6	219.8
100000	50	28	232.2	22.4	6.7	7.9	13.6	9.8	222.0
		Avg	218.2	28.9	4.4	5.5	6.3	5.7	213.2
10000	150	28	261.0	10.0	0.3	1.7	1.8	2.5	253.6
20000	150	28	275.8	18.4	1.0	2.6	2.5	4.0	264.4
30000	150	28	288.4	11.8	1.5	3.0	4.0	5.6	279
40000	150	28	295.2	9.0	2.0	3.7	6.7	7.0	287.8
50000	150	28	296.4	13.8	2.8	5.0	7.9	8.6	293.2
60000	150	28	304.0	9.6	3.8	5.7	9.9	9.2	297.2
70000	150	28	310.2	14.4	5.2	7.0	11.1	10.6	296.8
80000	150	28	314.2	19.2	6.7	8.7	13.0	12.0	301.4
90000	150	28	313.6	14.6	6.8	8.8	14.4	12.7	301.8
100000	150	28	314.6	12.8	7.6	9.5	19.6	15.1	313.8
		Avg	297.3	13.4	3.8	5.6	9.1	8.7	288.9
		Overall Avg	230.7	16.7	3.4	4.6	7.9	6.4	224.7

5. Conclusions. Benders decomposition applied to CDR-problems exhibits a novel structure in the feasible region of the dual slave problem, which has not been previously encountered in the literature. By analyzing this structure, we designed an integrated delayed Benders cut generation and simultaneous column-and-row generation algorithm to solve large-scale CDR-problems to optimality. An original feature of this algorithm is that the extreme points and extreme rays corresponding to the existing Benders cuts are lifted as part of the row-generating PSP in order to account for the increase in the dimension of the dual polyhedron. We provided an extensive analysis on the convergence of the proposed methodology and illustrated its mechanics by an application to the TCR problem. The computational experiments attest to the superior performance of our algorithm compared to solving the LP relaxation of TCR directly via `CPLEX`. Moreover, we also demonstrated that the new integrated delayed Benders cut generation and simultaneous column-and-row generation algorithm improves significantly over the direct application of simultaneous column-and-row generation (Muter et al., 2013a) to TCR.

In our current work, we apply column generation to both the master and the subproblems in a Benders decomposition framework instead of a single restricted master problem as in the column-and-row generation approach *CRG* by Muter et al. (2013a). That is, in this paper, we expose the decomposable structure of CDR-problems – which went unexploited in our earlier work – via Benders decomposition. Therefore, we expect our proposed approach *CRG-BDS* to be outperformed by *CRG* only if the Benders cuts are weak by nature for the specific problem at hand. In our illustrative application to the TCR problem, we show empirically that some instance types are inherently less suitable for Benders decomposition as they lead to slower convergence.

The end goal of our research agenda on CDR-problems is to embed our algorithms relying on simultaneous column-and-row generation into branch-and-price or branch-and-cut-and-price frameworks in order to solve large-scale CDR-problems with integrality restrictions. Equally important is the customization of the generic framework to specific CDR-problems giving rise to computationally viable implementations.

Acknowledgment This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grants 113M014 and 113M480. The authors would like to acknowledge the contribution of the COST Action TD1207.

Appendix A. Here are the three assumptions that were originally stipulated for CDR-problems in Muter et al. (2013a).

ASSUMPTION A.1 *The generation of a new set of variables* $\{y_k \mid k \in S_K\}$ *prompts the generation of a new set of variables* $\{x_n \mid n \in S_N(S_K)\}$. *Furthermore, a variable* $x_{n'}, n' \in S_N(S_K)$, *does not appear in any* linking constraints *other than those indexed by* $\Delta(S_K, S_N(S_K))$ *and introduced to the* (**SRMP**) *along with* $\{y_k \mid k \in S_K\}$ *and* $\{x_n \mid n \in S_N(S_K)\}$.

ASSUMPTION A.2 *A linking constraint is redundant until* all *variables in at least one of the minimal variable sets associated with this linking constraint are added to the* (**SRMP**).

ASSUMPTION A.3 *Suppose that we are given a minimal variable set* $\{y_l \mid l \in S_K\}$ *that generates a set of linking constraints* $\Delta(S_K)$ *and a set of associated* $x-$*variables* $\{x_n \mid n \in S_N(S_K)\}$. *When the set of linking constraints* $\Delta(S_K)$ *is* first *introduced into the* (**SRMP**) *during the column-and-row generation, then for each* $k \in S_K$ *there exists a constraint* $i \in \Delta(S_K)$ *of the form*

$$C_{ik} y_k + \sum_{n \in S_N(S_K)} D_{in} x_n \geq 0, \tag{38}$$

where $C_{ik} > 0$ *and* $D_{in} < 0$ *for all* $n \in S_N(S_K)$.

Appendix B. Here is the proof of Proposition 4.1.

PROOF. Suppose that the dual polyhedron is described as $\Phi = \{(\boldsymbol{v}, \boldsymbol{w}) \in \mathbb{R}^{|M|+|K|} \mid \begin{pmatrix} \mathbf{R}^v & \mathbf{R}^w \end{pmatrix} \begin{pmatrix} \boldsymbol{v} & \boldsymbol{w} \end{pmatrix}^\intercal \geq \mathbf{0}\}$ and $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}}) \in Q_\Phi$ is an extreme ray of unboundedness. Note that Φ is a polyhedral cone and by the definition of an extreme ray, $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ must lie on a 1-dimensional face of Φ. Therefore, $n-1$ linearly independent constraints are binding at $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ with $n = |M| + |K|$. Denoting the index set of the active constraints at $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ by E we have $\begin{pmatrix} \mathbf{R}^v_E & \mathbf{R}^w_E \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{v}} & \bar{\boldsymbol{w}} \end{pmatrix}^\intercal = \mathbf{0}$. The extreme ray $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ certifies (32) as unbounded only if $\begin{pmatrix} \mathbf{c}^v & \mathbf{c}^w \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{v}} & \bar{\boldsymbol{w}} \end{pmatrix}^\intercal < 0$ holds, where $\mathbf{c} \geq \mathbf{0}$ is the vector of objective function coefficients of (32). This condition implies that there exists $k \in K$ such that $\bar{w}_k < 0$ because $\mathbf{c} \geq \mathbf{0}$ and $\bar{\boldsymbol{v}} \geq \mathbf{0}$. This in turn requires the presence of $m \in M^k$ such that $\bar{v}_m > 0$ because $M^k \neq \emptyset$ and the feasibility of $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ for (32) prescribes $\bar{w}_k + \bar{v}_m \geq 0,\ m \in M^k$.

Without loss of generality, we can scale $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ such that $\bar{v}_m = 1$. Thus, $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ satisfies $\begin{pmatrix} \mathbf{R}^v_E & \mathbf{R}^w_E \\ 0 \cdots 1 \cdots 0 \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{v}} & \bar{\boldsymbol{w}} \end{pmatrix}^\intercal = \begin{pmatrix} \mathbf{R}^v_{E'} & \mathbf{R}^w_{E'} \end{pmatrix} \begin{pmatrix} \bar{\boldsymbol{v}} & \bar{\boldsymbol{w}} \end{pmatrix}^\intercal = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}$, where the single non-zero entry in the last row of $\mathbf{R}_{E'}$ is the coefficient of v_m. We conclude the proof by arguing that the rank of $\mathbf{R}_{E'}$ is n which establishes $(\bar{\boldsymbol{v}}, \bar{\boldsymbol{w}})$ as an extreme point of (35). Since the rank of $\mathbf{R}_E$ is $n-1$, the column associated with v_m in $\mathbf{R}_E$ can be expressed as a linear combination of the remaining $n-1$ linearly independent columns of $\mathbf{R}_E$. However, this is not true for $\mathbf{R}_{E'}$ because only v_m has a non-zero coefficient in the last row. Thus, the rank of $\mathbf{R}_{E'}$ is n. □

References

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.

Avella, P., D'Auria, B., and Salerno, S. (2006). A LP-based heuristic for a time-constrained routing problem. *European Journal of Operational Research*, 173(1):120–124.

Avella, P., Sassano, A., and Vasilev, I. (2007). Computational study of large-scale p-median problems. *Mathematical Programming*, 109(1):89–114.

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

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

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

Cordeau, J.-F., Soumis, F., and Desrosiers, J. (2000). A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34(2):133–149.

Cordeau, J.-F., Stojkovic, G., Soumis, F., and Desrosiers, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388.

Côté, J.-F., Dell'Amico, M., and Iori, M. (2014). Combinatorial Benders' cuts for the strip packing problem. *Operations Research*, 62(3):643–661.

Dantzig, G. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, 8(1):101–111.

Desaulniers, G., Desrosiers, J., and Spoorendonk, S. (2011). Cutting planes for branch-and-price algorithms. *Networks*, 58(4):301–310.

Desrosiers, J. and Lübbecke, M. E. (2011). Branch-Price-and-Cut Algorithms. In Cochran, J. J., editor, *Wiley Encyclopedia of Operations Research and Management Science (EORMS)*. John Wiley & Sons. http://dx.doi.org/10.1002/9780470400531.eorms0118.

Fischetti, M., Ljubić, I., and Sinnl, M. (2016). Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557–569.

Frangioni, A. and Gendron, B. (2009). 0-1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, 157(6):1229–1241.

Frangioni, A. and Gendron, B. (2013). A stabilized structured Dantzig-Wolfe decomposition method. *Mathematical Programming*, 140:45–76.

Gelareh, S., Monemi, R. N., and Nickel, S. (2015). Multi-period hub location problems in transportation. *Transportation Research Part E: Logistics and Transportation Review*, 75:67–94.

Katayama, N., Chen, M., and Kubo, M. (2009). A capacity scaling heuristic for the multicommodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232(1):90–101.

Lübbecke, M. E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

Mercier, A. (2008). A theoretical comparison of feasibility cuts for the integrated aircraft-routing and crew-pairing problem. *Transportation Science*, 42(1):87–104.

Muter, İ., Birbil, Ş.İ., and Bülbül, K. (2013a). Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. *Mathematical Programming*, 142(1-2):47–82.

Muter, İ., Birbil, Ş.İ., Bülbül, K., and Şahin, G. (2012). A note on "a LP-based heuristic for a time-constrained routing problem". *European Journal of Operational Research*, 221:306–307.

Muter, İ., Birbil, Ş.İ., Bülbül, K., Şahin, G., Taş, D., Tüzün, D., and Yenigün, H. (2013b). Solving a robust airline crew pairing problem with column generation. *Computers & Operations Research*, 40(3):815–830.

Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2016). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*.

Sadykov, R. and Vanderbeck, F. (2013). Column generation for extended formulations. *EURO Journal on Computational Optimization*, 1(1-2):81–115.

Wang, G. and Tang, L. (2010). A row-and-column generation method to a batch machine scheduling problem. In *Proceedings of the Ninth International Symposium on Operations Research and Its Applications (ISORA-10)*, pages 301–308, Chengdu-Jiuzhaigou, China.

Zak, E. (2002). Modeling multistage cutting stock problems. *European Journal of Operational Research*, 141(2):313–327.