

ISE



Industrial and
Systems Engineering

A Generalization of Benders' Algorithm for Two-Stage Stochastic Optimization Problems With Mixed Integer Recourse

TED K. RALPHS AND ANAHITA HASSANZADEH

Department of Industrial and Systems Engineering, Lehigh University, USA

COR@L Technical Report 14T-005



LEHIGH
UNIVERSITY.

COR@L
COMPUTATIONAL OPTIMIZATION
RESEARCH AT LEHIGH 

A Generalization of Benders' Algorithm for Two-Stage Stochastic Optimization Problems With Mixed Integer Recourse

TED K. RALPHS^{*1} AND ANAHITA HASSANZADEH^{†1}

¹Department of Industrial and Systems Engineering, Lehigh University, USA

August 3, 2014

Abstract

We describe a generalization of Benders' method for solving two-stage stochastic linear optimization problems in which there are both continuous and integer variables in the first and second stages. Benders' method relies on finding effective lower approximations for the value function of the second-stage problem. In this setting, the value function is a discontinuous, non-convex, piecewise polyhedral function in general. To obtain a convergent algorithm, we employ the strong dual functions encoded in the branch-and-bound trees resulting from solution of the second-stage problem. We show that these can be used effectively within Benders' framework and describe a method for obtaining all required dual functions from a single, continuously refined branch-and-bound tree that is used to warm start the solution procedure for each subproblem. Finally, we show that this procedure allows us to conclude there exists a single branch-and-bound tree that encodes the full value function.

1 Introduction

We consider the two-stage stochastic linear optimization problem given by

$$\min_{x \in \mathcal{P}_1} \Psi(x), \tag{SP}$$

where $\mathcal{P}_1 = \{x \in X \mid Ax = b\}$ is the *first-stage feasible region* defined by $A \in \mathbb{Q}^{m_1 \times n_1}$, $b \in \mathbb{Q}^{m_1}$, and $X = \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}$. The objective function Ψ is defined by

$$\Psi(x) = c^\top x + \Xi(x), \tag{1.1}$$

where $c \in \mathbb{R}^{n_1}$ reflects the immediate cost of implementation of the first-stage solution and Ξ is a risk measure reflecting the additional cost incurred as a result of uncertainty about the future. As is conventional, we take Ξ to be the expected cost of the *recourse problem*, henceforth referred to as the *second-stage problem*, which is a mixed integer linear optimization problem (MILP) parameterized on both the value of the first-stage solution and a random variable ω . We assume ω is drawn from

^{*}E-mail: ted@lehigh.edu

[†]E-mail: anahita.hassanzadeh@gmail.com

a given discrete and finite probability space (Ω, \mathcal{A}, P) so that ω represents which one of a finite number of explicitly enumerated scenarios is realized and p_ω represents the probability of such realization. Formally, the function Ξ is then defined by

$$\Xi(x) = \mathbb{E}_{\omega \in \Omega} [\phi(h_\omega - T_\omega x)] = \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x), \quad (1.2)$$

for $x \in \mathcal{P}_1$, where $T_\omega \in \mathbb{Q}^{m_2 \times n_1}$ and $h_\omega \in \mathbb{Q}^{m_2}$ represent the realized values of the stochastic inputs to the second stage for scenario $\omega \in \Omega$. The function ϕ is the *second-stage value function*, which encodes the cost of the recourse decision for a given first-stage solution x and realization ω . This value function plays a central role in the remainder of the paper, so we next define it formally and illustrate its structure through some examples that we refer to throughout the rest of the paper.

As indicated in (1.2), we assume that the second-stage problem is parameterized on both the first-stage solution and the realized stochastic inputs through joint determination of the right-hand side. Thus, for a given $\beta \in \mathbb{R}^{m_2}$, we have

$$\phi(\beta) = \min\{q^\top y \mid y \in \mathcal{P}_2(\beta)\}, \quad (\text{RV})$$

where $\mathcal{P}_2(\beta) = \{y \in Y \mid Wy = \beta\}$ is the *second-stage feasible region* with respect to β , defined by constraint matrix $W \in \mathbb{Q}^{m_2 \times n_2}$, the set $Y = \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2}$, and the second-stage objective function $q \in \mathbb{R}^{n_2}$, which represents the cost of recourse action. The following examples illustrate how the structure of the objective function Ψ relates to the structure of ϕ .

Example 1. Consider the following instance of a continuous two-stage problem ($r_1 = r_2 = 0$).

$$\begin{aligned} \min \quad & \Psi(x) = -3x_1 - 3.8x_2 + \sum_{\omega \in \Omega} 0.5\phi(h_\omega - 2x_1 - 0.5x_2), \\ \text{s.t.} \quad & x_1 \leq 5, \ x_2 \leq 5, \\ & x \in \mathbb{R}_+^2, \end{aligned} \quad (1.3)$$

where

$$\begin{aligned} \phi(\beta) = \min \quad & 6y_1 + 4y_2 + 3y_3 + 4y_4 + 5y_5 + 7y_6 \\ \text{s.t.} \quad & 2y_1 + 5y_2 - 2y_3 - 2y_4 + 5y_5 + 5y_6 = \beta, \\ & y \in \mathbb{R}_+^6, \end{aligned} \quad (1.4)$$

where $\Omega = \{1, 2\}$, $h_1 = 6, h_2 = 12$. Figures 1a and 1b show the form of the objective function Ψ and second-stage value function ϕ , respectively, for Example 1. In this case, both functions are convex and this is true in general for the continuous case. Note the similarities in shape of the two functions. The structure of Ψ clearly derives from that of ϕ . ■

Because ϕ is convex in the continuous case, Benders' method can be applied straightforwardly. This is true even with the introduction of integrality constraints in the first stage ($r_1 > 0$), as shown by Van Slyke and Wets (1969). Thus, when $r_2 = 0$, (SP) can be solved in principle using little more than a standard MILP solver. When $r_2 > 0$, ϕ is non-convex and discontinuous in general (Bank et al., 1983; Blair and Jeroslow, 1977, 1982; Hassanzadeh et al., 2014). In the next example, we illustrate the form of the value function in the pure integer case for which $r_2 = n_2$.

Example 2. Figure 2 shows two value functions resulting from the addition of integrality constraints

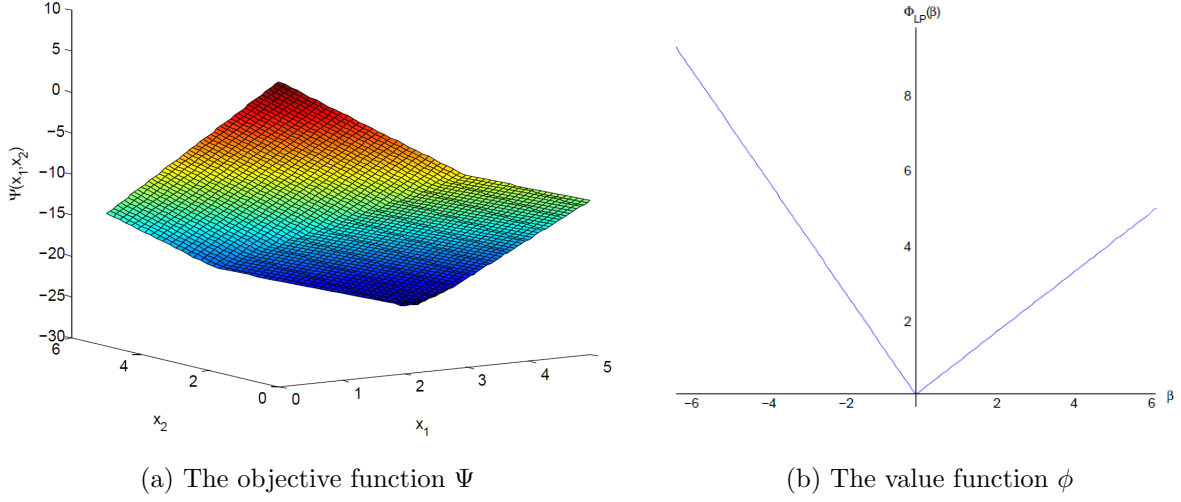


Figure 1: The objective function Ψ and second-stage value function ϕ in Example 1.

to the problem of Example 1 for all variables in the second stage. The points plotted in blue (closed circles) are the finite values of the value function of the resulting recourse problem, while the function in red (dashed lines and open circles) is the value function of the recourse problem when the single linear constraint is relaxed to the inequality $2y_1 + 5y_2 - 2y_3 - 2y_4 + 5y_5 + 5y_6 \leq \beta$. ■

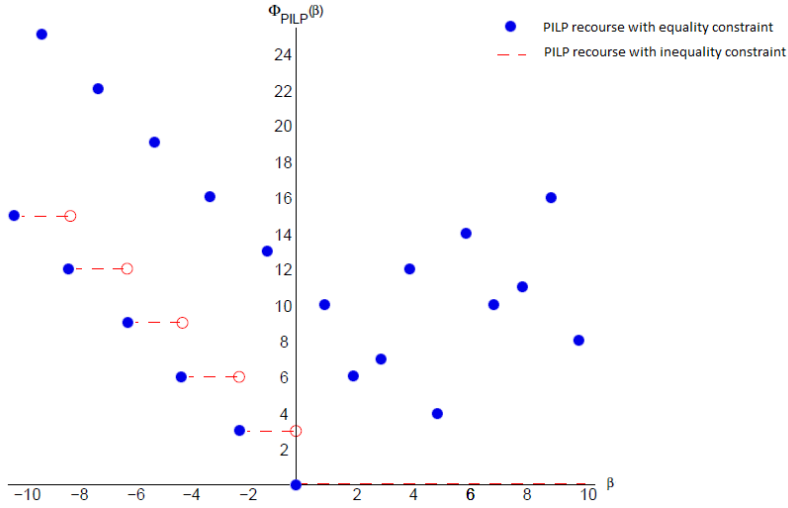


Figure 2: The value functions of two pure integer variations of (1.4)

In the pure integer case, the discrete nature of the problem is evident in the structure of the value function, which is only finite on a discrete set of points. In the inequality form, the value function remains constant over a countable number of regions of the domain. The discrete structure of the value function in this special case has been exploited in the development of several solution

methods relying on combinatorial enumeration schemes (Ahmed et al., 2004; Kong et al., 2006; Trapp et al., 2013; Schultz et al., 1998). The structure above changes substantially when we have both continuous and integer variables in the second-stage problem. Nevertheless, the discrete structure of the underlying *pure integer restriction* remains and we will show in Theorem 1 how to generalize the properties evident in the pure integer case to this more general case. We illustrate the structure of the value function in the general case in the next example.

Example 3. Consider the mixed integer variation of Example 1 where

$$\begin{aligned} \phi(\beta) = \min & 6y_1 + 4y_2 + 3y_3 + 4y_4 + 5y_5 + 7y_6 \\ \text{s.t.} & 2y_1 + 5y_2 - 2y_3 - 2y_4 + 5y_5 + 5y_6 = \beta \\ & y_1, y_2, y_3 \in \mathbb{Z}_+, y_4, y_5, y_6 \in \mathbb{R}_+. \end{aligned} \quad (1.5)$$

Figures 3a and 3b respectively show the objective function and second-stage value function, respectively, for this mixed integer variation of the problem from Example 1. Note once again the

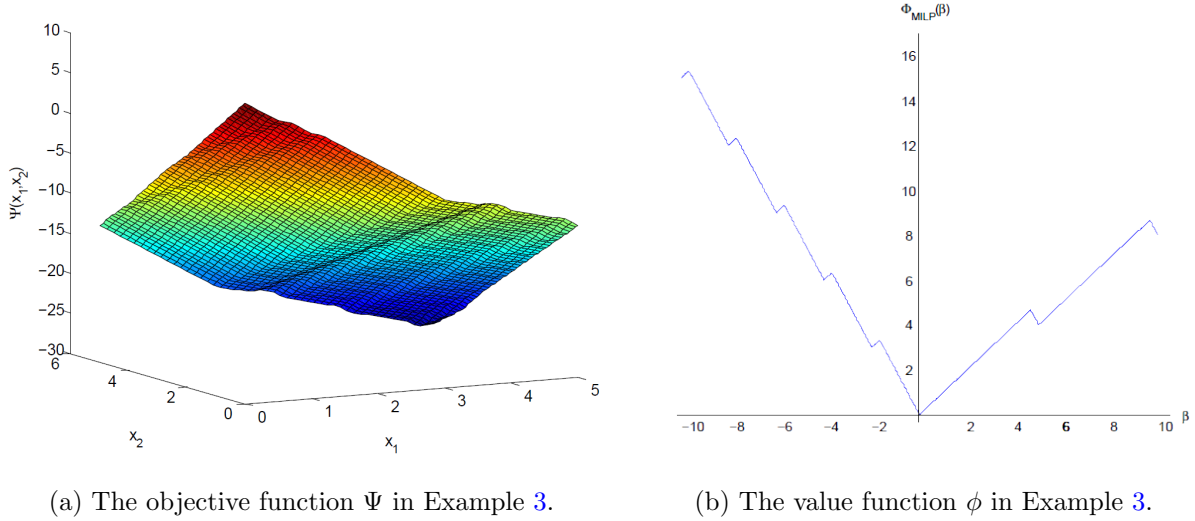


Figure 3: The objective and second-stage value functions for Example 3.

similarity in structure of the two functions. ■

The classical Benders' method utilizes subgradient information to generate a piecewise polyhedral convex approximation of the second-stage value function and thus only applies to the case of a convex value function. As Example 3 should make clear, solution methods for the more general non-convex case must either exploit special structure, such as in Example 2, or rely on a more general class of functions for approximating the value function from below. Most previous work has taken the first approach. Here, we take the second. For the remainder of the paper, we assume the following:

A1 $\Xi(x) = \mathbb{E}_{\omega \in \Omega} [\phi(h_\omega - T_\omega x)]$ is finite for all $x \in \mathcal{P}_1$.

A2 \mathcal{P}_1 is compact.

A3 The random variable ω is drawn from a discrete distribution with finite support.

Assumptions A1–A3 are not restrictive and are common in the literature. In Assumption A1, $\Xi[x] < +\infty$ requires the feasibility of the recourse problem for any right-hand side $h_\omega - T_\omega x$, $\omega \in \Omega$ with a given $x \in \mathcal{P}_1$. This is known as the *relative recourse* property and can be guaranteed by adding artificial variables to the recourse problem. By $\Xi[x] > -\infty$, we require the dual polyhedron of the linear relaxation of the recourse problem to be nonempty. We make Assumption A2 to guarantee the finite termination of our algorithm (this is not required to use the algorithm in practice).

Assumption A3 assures the expectation in (SP) can be expressed as the sum of a finite number of terms, which allows for reformulation of (SP) as a large-scale MILP, the so-called *deterministic equivalent* problem:

$$\begin{aligned} \min \quad & c^\top x + \sum_{\omega \in \Omega} p_\omega q^\top y_\omega \\ \text{s.t.} \quad & W y_\omega = h_\omega - T_\omega x \quad \forall \omega \in \Omega \\ & x \in \mathcal{P}_1, y \in Y. \end{aligned} \tag{DE}$$

For a fixed x , (SP) can be separated into $|\Omega|$ independent subproblems, so applying a decomposition method directly to this reformulation is one possible solution approach. On the surface, such a decomposition by scenario makes sense, but one argument against this tactic is that with a large number of scenarios, there may be substantial overlap in the areas of the value function that are relevant in each scenario. By taking a global view in maintaining one approximation of the value function, we hope to avoid redundant enumeration.

The remainder of the paper is organized as follows. In Section 2, we review the literature on two-stage stochastic problem with integer recourse. In Section 3, we describe our discrete representation of the MILP value function and use this to propose a deterministic reformulation of (SP). In Section 4, we introduce the approximating functions we derive from the branch-and-bound tree when solving the subproblems that arise in the course of the algorithm. We show that all the approximating functions required to solve the two-stage problem can be obtained from a single tree by applying principles of warm-starting for integer optimization. We further derive that there exists a single tree that encodes the full MILP value function. Section 5 contains the framework of the generalization of Benders’ algorithm. We discuss the use of branch-and-bound approximating function within the general framework and provide a formulation for the resulting Benders’ master problem as a MILP. Furthermore, we show that the proposed algorithm is convergent and correct. Section 7 includes the summary of this work and remarks for future research.

2 Related Work

To date, the majority of the work done on solution of two-stage stochastic linear optimization problems has been on the case of a (mixed) binary or pure integer second-stage problem. Table 2.1 provides a summary of the methods proposed to date and the assumptions required for the employment of each method. The first two sets of columns specify the assumptions made on integrality of variables, while the third set of columns describes the stochasticity in the input. Below, we briefly review the algorithms in this table, as well as other related work.

It is natural that various special cases involving binary variables have received the most attention, given the rich theory that has been developed specifically addressing this case. In the early

work of Carøe and Tind (1997), the authors suggested the use of disjunctive programming and lift-and-project cuts when the recourse is mixed binary. They first decomposed the feasible set of (DE) into $|\Omega|$ subsets, which they sequentially convexified to generate the convex hull of each subset. Sherali and Fraticelli (2002) modified Benders' method by generating valid inequalities in the subproblems. Assuming the first-stage variables are binary, each such valid inequality can be generated using the reformulation linearization technique (RLT) or lift-and-project. To ensure the generated valid inequality is globally valid, it can be re-expressed as a function of the first-stage variables. Using the dual solution of the optimal subproblem, optimality cuts for the master problem can be generated. In the same vein, Sen and Hingle (2005) developed valid inequalities in both stages. The valid inequalities to augment the linear relaxation of the second-stage problem were generated so as to be valid for the union of disjunctive sets obtained by a disjunction arising from a fractional second-stage variable. Ntaimo (2010) provided a variation of the latter for problems with fixed T and h . Sherali and Zhu (2006) extended the framework of Sherali and Fraticelli (2002) to accommodate binary variables in the first stage by using decomposition and global branch-and-bound methods. Sherali and Smith (2009) used the RLT to devise a specialization of Benders' algorithm for two-stage stochastic risk management problems with a pure binary first-stage problem.

For the case of a pure integer second-stage problem, Gade et al. (2012) recently used Benders' decomposition and generated valid inequalities in both stages. The proposed method solves the second-stage subproblems as LPs and iteratively adds Gomory valid inequalities to the description of the subproblems. The generated valid inequalities are parametrized as a function of the first-stage variables. Taking a different approach, Schultz et al. (1998) characterized regions of the right-hand side over which the second-stage value function is constant. This allows for a countable partition on the set \mathcal{P}_1 using which, one can determine the corresponding level-sets of the objective function. The candidate points from these level-sets are enumerated and evaluated using the Gröbner basis of the recourse pure integer linear problem (PILP). Building on results on level sets of a pure integer value function, Ahmed et al. (2004) proposed the value function reformulation of (SP). Through a variable transformation, a global branch-and-bound algorithm was applied to the problem and was shown to be finite. This work assumed general first-stage variables and a fixed matrix T . Restricting the stochasticity to the right-hand-side vector h , Kong et al. (2006) proposed a procedure for finite construction of the value function when \mathcal{P}_1 is finite. Furthermore, combining the results of Schultz et al. (1998) on level sets and the value function reformulation of Ahmed et al. (2004), the authors provided a characterization of certain candidate points, the so-called *minimal tenders*, in partitioning the right-hand side region. Trapp et al. (2013) proposed an alternative global branch-and-bound method that optimizes over a certain integral monoid. Finiteness of all algorithms in this category rely on pure integrality in the second-stage problem.

The first work to consider general recourse problems (those with both continuous and integer variables) was due to Laporte and Louveaux (1993). They proposed a modification of Benders' method that requires the solution of second-stage subproblems, as in the classical method, but the optimality cut used was a specialized linear cut that is valid only for problems with binary first-stage variables. Carøe and Tind (1998) pioneered the use of integer programming duality theory to develop optimality cuts for the Benders' framework, but their algorithm was designed for problems with pure integer recourse. They demonstrated how dual functions generated from a cutting-plane method with Gomory cuts can be used to generate optimality cuts. In the disjunctive decomposition branch-and-bound method of Sen and Sherali (2006), the second-stage problems were general and were solved with a branch-and-bound algorithm. The dual function obtained from each tree was

	First Stage			Second Stage			Stochasticity			
	\mathbb{R}	\mathbb{Z}	\mathbb{B}	\mathbb{R}	\mathbb{Z}	\mathbb{B}	\mathbf{W}	\mathbf{T}	\mathbf{h}	\mathbf{q}
Laporte and Louveaux (1993)			*	*	*	*	*	*	*	
Carøe and Tind (1997)	*		*	*		*	*	*	*	*
Carøe and Tind (1998)	*	*	*		*	*		*	*	
Carøe and Schultz (1998)	*	*	*	*	*	*		*	*	*
Schultz et al. (1998)	*				*	*			*	
Sherali and Fraticelli (2002)			*	*		*	*	*	*	*
Ahmed et al. (2004)	*	*	*		*	*	*		*	*
Sen and Higle (2005)			*	*		*		*	*	
Sen and Sherali (2006)			*	*	*	*		*	*	
Sherali and Zhu (2006)	*		*	*		*	*	*	*	
Kong et al. (2006)		*	*		*	*	*	*	*	*
Sherali and Smith (2009)			*	*		*	*	*	*	*
Yuan and Sen (2009)			*	*		*		*	*	*
Ntaimo (2010)			*	*		*	*			*
Gade et al. (2012)			*		*	*	*	*	*	*
Trapp et al. (2013)		*	*		*	*			*	
Current work	*	*	*	*	*	*		*	*	

Table 2.1: Assumptions made in related algorithms

modified to derive a valid inequality for the first-stage problem. Finally, a convexification technique similar to the one of Sen and Higle (2005) was used to linearize these inequalities. The disjunctive convexification technique used requires binary first-stage variables. Computational improvements to cut generation in the latter work were reported in Yuan and Sen (2009). Carøe and Schultz (1998) accommodated general variables in both stages through a fundamentally different approach called *dual decomposition*, which relaxed the so-called *non-anticipativity* constraint in a Lagrangian fashion. Carøe and Tind (1998) suggested the use of dual functions as optimality cuts in a fashion similar to what we describe herein. Sen and Sherali (2006) solved the subproblems from each scenario as a generic MILP and obtained dual functions from the branch-and-bound tree, as we do, but then convexified them, which again restricts the form of the first-stage problem.

3 Value Function Reformulation

In this section, we discuss the structure of the value function of a MILP and its importance in solving (SP). This material is largely based on results of Hassanzadeh et al. (2014). Although we have endeavored to make this paper self-contained, a number of details from the previous paper have been left out. In particular, Theorem 1 is stated below without proof because the development is quite lengthy. We refer the interested reader to the aforementioned paper for background.

In Hassanzadeh et al. (2014), we propose a characterization of the value function that we now show leads to a deterministic reformulation of (SP) that is distinct from (DE). Although much of what we discuss here is true for general value functions, we use our notation for the second-stage problem throughout and also define some additional notation for convenience. Let $I = \{1, \dots, r_2\}$, $C = \{r_2 + 1, \dots, n_2\}$, and $N = I \cup C$ be the relevant sets of indices of second-stage variables. For

any index set $D \subseteq N$ and any vector z indexed on N , we denote by z_D the sub-vector consisting of the corresponding components of z . Similarly, for a matrix $M \in \mathbb{Q}^{m \times n}$, we denote by M_D the sub-matrix constructed by columns of M that correspond to indices in D . We will also use this notation to refer to a given sub-vector, e.g., $y_I \in \mathbb{Z}^{r_2}$, of a vector $y \in Y$ for which the remaining components have not been specified.

One of the major themes of [Hassanzadeh et al. \(2014\)](#) is that the structure of the value function of a MILP arises from the structure of two related functions:

$$\phi_C(d) = \min_{y_C \in \mathbb{R}_+^{n_2 - r_2}} \{q_C^\top y_C \mid W_C y_C = d\}, \quad (3.1)$$

which is the value function of the *continuous restriction*, and

$$\phi_I(d) = \min_{y_I \in \mathbb{Z}_+^{r_2}} \{q_I^\top y_I \mid W_I y_I = d\}, \quad (3.2)$$

the value function of the similarly defined *integer restriction*. The second-stage value function (RV) can be re-written as a combination of these two related value functions in the following way

$$\phi(\beta) = \min_{y_I \in \mathbb{Z}_+^{r_2}} \{\phi_I(W_I y_I) + \phi_C(\beta - W_I y_I)\} = \min_{y_I \in \mathbb{Z}_+^{r_2}} \{q_I^\top y_I + \phi_C(\beta - W_I y_I)\}, \quad (3.3)$$

Even if we have a complete description of ϕ_C , this characterization of the value function would not be practical, since we obviously cannot enumerate the set $\mathbb{Z}_+^{r_2}$. Fortunately, though, there exists a set \mathcal{S} that is (1) finite under mild assumptions, (2) can be constructed algorithmically, and (3) is necessary and sufficient to describe ϕ . The following result states this formally.

Theorem 1 ([Hassanzadeh et al., 2014](#)) *Let B_{SLC} be the set of points of strict local convexity of ϕ . Then there exists $\mathcal{S} \subseteq \mathbb{Z}_+^{r_2}$ such that $B_{SLC} = \{W_I y_I \mid y_I \in \mathcal{S}\}$. Furthermore, if $\{\beta \in \mathbb{R}^{m_2} \mid \phi_I(\beta) < \infty\}$ is finite and \mathcal{S} is a minimal such set, then \mathcal{S} is finite and*

$$\phi(\beta) = \min_{y_I \in \mathcal{S}} \{q_I^\top y_I + \phi_C(\beta - W_I y_I)\} \quad \forall \beta \in \mathbb{R}^{m_2}. \quad (3.4)$$

This result guarantees that points in \mathcal{S} are solutions to the pure integer restriction, so one consequence is that, roughly speaking, we only need to consider solutions of this pure integer linear optimization problem when constructing the value function of a general MILP. Furthermore, we only need to consider a small subset of those solutions to get the full value function. The notion of strict local convexity that we utilize generalizes the notion of minimal tenders from ([Trapp et al., 2013](#)). Each such point is the minimizer over an associated region for which the value function is convex. Over such regions, the integer part of the optimal solution remains constant and the value function of the MILP is simply a translation of the value function of the continuous restriction to that point. These regions are a generalization of the regions described by [Schultz et al. \(1998\)](#) over which the value function remains constant in the pure integer case. Figure 5, which appears later in the paper, shows these regions for the second-stage problem from Example 3.

In ([Hassanzadeh et al., 2014](#)), we describe an algorithm for generating a superset of \mathcal{S} that can be implemented in practice. This result gives us a way of generating a complete description of ϕ that can be embedded in (SP) to obtain the aforementioned deterministic reformulation. Note that in this context, we are only really interested in parts of the domain that can actually arise from

first-stage solutions in some scenario, e.g., we only need to know the value function over the set $B = \cup_{\omega \in \Omega} \{h_\omega - T_\omega x \mid x \in \mathcal{P}_1\}$, which is bounded by our assumptions. We do not depend on the finiteness of \mathcal{S} , but on that of the set

$$\mathcal{B} = \mathcal{S} \cap \text{proj}_{\mathbb{Z}^{r_2}} (\cup_{b \in B} \mathcal{P}_2(b)). \quad (3.5)$$

To derive the reformulation, we first rewrite (SP) as

$$\min_{x \in \mathcal{P}_1} c^\top x + \sum_{\omega \in \Omega} p_\omega \left(\min_{y_I \in \mathcal{B}} \left\{ q_I^\top y_I + \phi_C(h_\omega - T_\omega x - W_I y_I) \right\} \right). \quad (3.6)$$

using Theorem 1. From assumption A1, we have that the dual polyhedron of the linear relaxation of the recourse problem is always nonempty. Let us denote this polyhedron by \mathcal{D} . From LP duality, we have that $\phi_C(d) = \sup_{z \in \mathcal{D}} d^\top z$ for $d \in \mathbb{R}^{m_2}$. Let $\{\nu^i\}_{i \in K}$ be the set of extreme points of \mathcal{D} indexed by set K . When \mathcal{D} is unbounded, let $\{\sigma^j\}_{j \in L}$ be its set of extreme directions indexed by set L . For $d \in \mathbb{R}^{m_2}$, if $\phi_C(d) < \infty$, then

$$\phi_C(d) = \max_{i \in K} d^\top \nu^i. \quad (3.7)$$

Otherwise, for some $j \in L$, we have $d^\top \sigma^j > 0$ and $\phi_C(d) = +\infty$. In principle, one can generate the set of extreme points of a bounded polyhedron via a vertex enumeration method, such as the method of Avis and Fukuda (1992). The same method can be used to obtain extreme rays by bounding the polyhedron artificially. This allows us to rewrite (3.6) as a deterministic MILP. We let the set \mathcal{B} be indexed by set S .

$$\begin{aligned} \min \quad & c^\top x + \sum_{\omega \in \Omega} p_\omega \gamma(x, \omega) \\ \text{s.t.} \quad & \gamma(x, \omega) \geq q_I^\top y_I^s + \zeta(x, \omega) - M^{s, \omega} (1 - u^{s, \omega}) \quad \forall s \in S, \omega \in \Omega \\ & \zeta(x, \omega) \geq (h_\omega - T_\omega x - W_I y_I^s)^\top \nu^i \quad \forall i \in K, s \in S, \omega \in \Omega \\ & 0 \leq (h_\omega - T_\omega x - W_I y_I^s)^\top \sigma^j \quad \forall j \in L, s \in S, \omega \in \Omega \\ & \sum_{s \in S} u^{s, \omega} = 1 \quad \forall \omega \in \Omega \\ & x \in \mathcal{P}_1 \\ & u^{s, \omega} \in \mathbb{B}. \quad \forall s \in S, \omega \in \Omega \end{aligned} \quad (\text{VF})$$

where $M^{s, \omega} \geq \max_{x \in \mathcal{P}_1} \{|(h_\omega - T_\omega x - W_I y_I^s)^\top \nu^i|\}$ for all $\omega \in \Omega$, $i \in K$ and $s \in S$. Here, $\gamma(x, \omega)$ represents the cost of the recourse for a given scenario and first-stage solution. In the first constraint, this variable is used to represent the value function of the second-stage problem as described in (3.4), by minimizing over the set \mathcal{B} . This constraint, together with the fourth constraint, guarantees that equality holds for at least one of the constraints associated with $y_I \in \mathcal{B}$ and holds for that which ϕ achieves the minimum. The variable $\zeta(x, \omega)$ is the cost of the continuous restriction problem with respect to the fixed integer vector y_I , i.e., $\zeta(x, \omega) = \phi_C(h_\omega - T_\omega x - W_I y_I)$. The second and third constraints represent ϕ_C in terms of the extreme points and rays of the associated dual problem, as indicated in (3.7).

Generating the formulation (VF) requires generation of \mathcal{B} , K , and L a priori. Although these operations would clearly be expensive, they are nevertheless finite under our assumptions. Therefore, this reformulation yields a finite algorithm that can be seen as an alternative to solution of

the deterministic equivalent. Although a complete description of the value function is embedded within (VF), its size is independent of the size of the formulation of the second-stage problem and depends only on the number of scenarios and the cardinalities of \mathcal{B} , K and L . This reformulation could be smaller than (DE) in cases where ϕ has a simple structure and the cardinalities of K and L are relatively small. In practice, however, these sets can be large and solving (VF) directly would be cumbersome in many cases. An obvious idea for overcoming this complexity, which is the basis for our method, would be to generate parts of the formulation dynamically. We describe how to do that in the next section.

4 Lower Approximation of the Value Function

As a first step toward a complete specification of the generalization of Benders' method which is the main topic of this paper, we now describe the details of the most critical component, a procedure for dynamically constructing lower approximations of the value function from information generated by solution of instances of the second-stage problem. We describe in Section 5 how the procedure can be embedded within Benders' framework and tightly coupled with the procedure for solving the master problem. However, it is useful to first view the method simply as a technique for dynamically constructing the value function within a single continuously refined branch-and-bound tree.

The conceptual framework we present here is based upon the theory of duality and associated procedures for constructing so-called *dual functions* (for an overview of this topic, see (Güzelsoy and Ralphs, 2007; Güzelsoy, 2009)). Formally, a dual function is simply a function that bounds the value function from below, though the dual functions that arise in practice are by-products of particular solution algorithms and thus typically have special structure. We restrict our definition to the class of optimization problems considered here, though the concepts in this section are more general.

Definition 1 A function $f : \mathbb{R}^{m_2} \rightarrow \mathbb{R} \cup \{\pm\infty\}$ is said to be dual to the value function ϕ if

$$f(\beta) \leq \phi(\beta) \quad \forall \beta \in \mathbb{R}^{m_2}. \quad (4.1)$$

f is strong at $\hat{\beta} \in \mathbb{R}^{m_2}$ if $f(\hat{\beta}) = \phi(\hat{\beta})$.

Strong dual functions are important in optimization for several reasons. First, they provide proofs of optimality—most solution algorithms can be seen as methods for constructing such proofs. Second, they provide a natural way of performing sensitivity analyses, since such a function provides provable bounds on the optimal solution value for modified instances. Third, they provide methods of warm-starting the solution process, since the function produced when evaluating $\phi(\hat{\beta})$ for some $\hat{\beta} \in \mathbb{R}^{m_2}$ is likely similar or even identical to that produced when evaluating ϕ in a close neighborhood of $\hat{\beta}$.

Dual functions can be obtained within a number of different primal solution frameworks, but the two primary such frameworks are the cutting-plane method and the branch-and-bound method. The Gomory cutting plane algorithm has a natural interpretation as a method of constructing function in the class known as *Chvátal functions*. These are obtained as compositions of simple functions mirroring the sequences of operations (rounding and weighted combination) that are applied to produce Gomory cuts. Blair and Jeroslow (1982) showed that the value function of a pure integer program could be constructed in a finite number of steps by taking the maximum of

a finite number of such functions. Such a method does not seem practical, however, since it is well-known to suffer from numerical instability.

Dual functions arising from branch and bound (and also potentially from branch and cut) appear to be more practical. [Wolsey \(1981\)](#) was the first to propose deriving such dual functions. The idea is encapsulated in the following result.

Theorem 2 ([Wolsey, 1981](#)) *Let $\hat{\beta} \in \mathbb{R}^{m_2}$ be such that $\phi(\hat{\beta}) < \infty$ and suppose T indexes the set of terminating nodes of a branch-and-bound tree resulting from evaluation of $\phi(\hat{\beta})$. Then there exists a function $\underline{\phi}(\beta)$ dual to ϕ defined by*

$$\underline{\phi}(\beta) = \min_{t \in T} (\beta^\top \pi^t + \alpha^t) \quad \forall \beta \in \mathbb{R}^{m_2}, \quad (4.2)$$

where $\pi^t \in \mathbb{R}^{m_2}$ and $\alpha^t \in \mathbb{R}$ are feasible solutions to the dual of the LP relaxation associated with node t and for which $\underline{\phi}(\hat{\beta}) = \phi(\hat{\beta})$.

Proof. Consider evaluating $\phi(\hat{\beta})$ by a branch-and-bound procedure and let $t \in T$ be the index of a given node in the final branch-and-bound tree. The bound yielded by the associated subproblem is obtained by evaluation of the value function of its LP relaxation, which is of the form

$$\phi^t(\beta) = \min \{q^\top y \mid y \in \mathcal{P}_2^t(\beta)\}, \quad (P^t)$$

where

$$\mathcal{P}_2^t(\beta) = \{y \in \mathbb{R}_+^{n_2} \mid Wy = \beta, l_t \leq y \leq u_t\} \quad (4.3)$$

and $l_t, u_t \in \mathbb{Z}_+^{n_2}$ are the branching bounds applied to the variables (the upper bounds may be $+\infty$). The LP dual of (P^t) is

$$\max \{\hat{\beta}^\top \pi^t + l_t^\top \underline{\pi}^t - u_t^\top \bar{\pi}^t \mid (\pi^t, \underline{\pi}^t, \bar{\pi}^t) \in \mathcal{D}\}, \quad (D^t)$$

where π^t represents the dual variables associated with the matrix W from the original formulation and $\underline{\pi}^t, \bar{\pi}^t$ are the dual variable associated with the lower and upper bound constraints, respectively. The feasible set of the dual is

$$\mathcal{D} = \{(\pi^t, \underline{\pi}^t, \bar{\pi}^t) \in \mathbb{R}^{m_2} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_2} \mid W^\top \pi^t + \underline{\pi}^t - \bar{\pi}^t \leq q, \underline{\pi}^t \geq 0, \bar{\pi}^t \geq 0\}. \quad (4.4)$$

The dual problem only depends on $\hat{\beta}$ through its objective function, so the set \mathcal{D} is independent of $\hat{\beta}$. By LP duality, we then have that for any $(\pi^t, \underline{\pi}^t, \bar{\pi}^t) \in \mathcal{D}$ and $\beta \in \mathbb{R}^{m_2}$,

$$\beta^\top \pi^t + l_t^\top \underline{\pi}^t - u_t^\top \bar{\pi}^t \leq q^\top y. \quad (4.5)$$

Now, we consider two cases.

1. If $\phi^t(\hat{\beta}) < \infty$, then letting $(\hat{\pi}^t, \hat{\underline{\pi}}^t, \hat{\bar{\pi}}^t)$ be the optimal solution to (D^t) , we have that

$$\underline{\phi}^t(\beta) = \beta^\top \hat{\pi}^t + l_t^\top \hat{\underline{\pi}}^t - u_t^\top \hat{\bar{\pi}}^t = \beta^\top \hat{\pi}^t + \alpha^t \quad (4.6)$$

is dual to ϕ^t .

2. If $\phi^t(\hat{\beta}) = \infty$, we let $(\hat{\pi}^t, \hat{\underline{\pi}}^t, \hat{\bar{\pi}}^t)$ be any member of \mathcal{D} such that

$$\beta^\top \hat{\pi}^t + l_t^\top \hat{\underline{\pi}}^t - u_t^\top \hat{\bar{\pi}}^t > \phi(\hat{\beta}). \quad (4.7)$$

Such a member of \mathcal{D} must exist since $\phi^t(\hat{\beta}) = \infty$. Then by the same argument as we gave previously,

$$\beta^\top \hat{\pi}^t + l_t^\top \hat{\underline{\pi}}^t - u_t^\top \hat{\bar{\pi}}^t = \beta^\top \hat{\pi}^t + \alpha^t \quad (4.8)$$

is dual to ϕ^t .

Finally, by taking the minimum over the set of dual functions for the individual nodes, we obtain the function

$$\underline{\phi}(\beta) = \min_{t \in T} \underline{\phi}^t(\beta) \leq \phi(\beta), \quad (4.9)$$

which is dual to ϕ and has the form posited in the statement of the theorem.

It remains to show that $\underline{\phi}$ is strong. To see this, note that there must exist a node $t^* \in T$ such that $\underline{\phi}^{t^*}(\hat{\beta}) = \phi(\hat{\beta})$. Furthermore, we must have that $\underline{\phi}^t(\hat{\beta}) \geq \phi(\hat{\beta}) \forall t \in T$, since for each $t \in T$, we have either that (P^t) is feasible (in which case $\underline{\phi}^t(\hat{\beta}) \geq \phi(\hat{\beta})$ by the optimality of the branch-and-bound tree) or (P^t) is infeasible (in which case $\underline{\phi}^t(\hat{\beta}) \geq \phi(\hat{\beta})$ by construction). ■

The interpretation of the function $\underline{\phi}$ above is conceptually straightforward. The solution to the LP relaxation (P^t) of node t in the branch-and-bound tree yields the standard LP dual function, which bounds the optimal value of that subproblem. The overall lower bound yielded by the tree is the smallest bound yielded by any of the leaf nodes. This is the usual lower bound yielded by a branch-and-bound-based MILP solver during the solution process. By interpreting the optimal solution to the dual of the LP relaxation in each node as a *function*, we obtain $\underline{\phi}$.

One subtle point we should address further has to do with the infeasible nodes. In the proof above, we simply appealed to the existence of a dual solution that could be used to construct an appropriate dual function. In practice, we need to be able to obtain such dual solution in a practical way. Consider again a node $t \in T$ for which $\phi^t(\hat{\beta}) = \infty$. One method of obtaining an appropriate dual solution is to let $(\sigma^t, \underline{\sigma}^t, \bar{\sigma}^t)$ be an extreme ray of \mathcal{D} that proves the infeasibility of (P^t) (produced by the simplex algorithm used to solve the LP) and let $(\tilde{\pi}^t, \tilde{\underline{\pi}}^t, \tilde{\bar{\pi}}^t)$ be the member of \mathcal{D} generated just prior to discovery of the dual ray. By adding an appropriately chosen scalar multiple of the ray to this dual solution, we obtain a second dual solution with the desired property. More formally, let $\lambda \in \mathbb{R}^+$ be a given scalar and consider

$$(\hat{\pi}^t, \hat{\underline{\pi}}^t, \hat{\bar{\pi}}^t) = (\tilde{\pi}^t, \tilde{\underline{\pi}}^t, \tilde{\bar{\pi}}^t) + \lambda(\sigma^t, \underline{\sigma}^t, \bar{\sigma}^t) \quad (4.10)$$

By choosing λ large enough, we obtain a solution appropriate for use in (4.8). In practice, we may also avoid this issue by putting an explicit bound on the dual objective function value, since once the objective value of the current dual solution exceeds the global upper bound, the solution method can be terminated. In this case, the dual solution generated in the last iteration would itself be a solution that has the required property.

The function $\underline{\phi}$ associated with a given branch-and-bound tree is a concave piecewise polyhedral function whose gradients are the optimal dual solutions to the LP relaxations of the leaf nodes with respect to a fixed right-hand side $\hat{\beta}$. Theorem 2 provides a recipe for producing a tree that

corresponds to a strong dual function for a given $\beta \in \mathbb{R}^{m_2}$. This is already enough to allow us to state a convergent generalization of Benders' method in which we produce a different such function each time we solve the second-stage problem. In essence, this would mean constructing a new dual function for each scenario in each iteration of the algorithm. This approach would work in principle, but in practice, there may be substantial overlap between the dual functions produced and the resulting master problem may be much bigger than necessary. Furthermore, starting each subproblem solve from scratch will result in many repeated computations. We can improve on this basic framework by incorporating *warm-starting* techniques.

Warm starting procedures in integer linear optimization have been previously studied and implemented in the mixed integer optimization solver SYMPHONY (Ralphs and Güzelsoy, 2005, 2006) to accelerate solution of new problem instances by utilizing information obtained from solution of a base instance. The process is initialized by collecting the optimal bases of the current tree's terminating nodes, as described in the proof of Theorem 2. These previous optimal bases are used to warm-start solution of the LP relaxations in each leaf node, producing a set of candidate terminating nodes that require further branching (due to the introduction of fractional value for some of the integer variables). After the initial phase, branching is continued as usual until a tree that is optimal with respect to the new right-hand side is found. We now illustrate the concept of warm-starting the solution process by showing how we would solve a sequence of subproblems within Benders' method, warm starting the solution of each subproblem from the tree generated by solution of the previous subproblem.

Example 4. Although we have not formally stated our generalization of Benders' method yet, it should be clear that the method consists of the iterative solution of a sequence of subproblems with different right-hand sides. The goal of each step in this process is to improve the approximation of the value function in an area where it is currently not strong in response to the proposed first-stage solution, generated with respect to that current approximation.

Consider now the second-stage problem from Example 3. In Figure 4, we show the steps in solving this second-stage problem for four right-hand sides, 5.5, 11.5, 4 and 10, arising from execution of Benders' method. Consider first the value function of the LP relaxation of the original problem. Over the interval $(0, 12)$, this function is $g_0 = 0.8\beta$ and is strong at right-hand sides 0, 5 and 10. For these right-hand sides, the solution to the LP relaxation is integer-valued and the branch-and-bound tree would consist of a single node.

The function g_0 lies strictly below the value function at $\beta = 5.5$, which indicates that for this right-hand side, one of the integer variables (y_2) takes on a fractional value. We therefore branch on y_2 to produce two affine functions, g_1 and g_2 . The resulting dual function, $\min\{g_1, g_2\}$, is a piecewise concave dual function and is strong at $\beta = 5.5$.

In evaluating $\phi(11.5)$, we begin with the optimal tree from the previous iteration. Re-optimizing the LPs at nodes 1 and node 2 with the new right-hand side, we obtain a new dual solution for node 2, while the function of node 1 is already optimal for $\beta = 11.5$ and the primal solution at node 1 is still integer feasible. Node 2 produces a primal solution in which y_2 is fractional and we thus have to branch on y_2 *again*. Note that this situation cannot arise in branch-and-bound for a single fixed instance, since the value of the variable that was just branched on must be integer in the resulting subproblems. After branching, we obtain two new affine functions.

For $\beta = 4$, we similarly re-optimize the LP relaxations at nodes 1, 3 and 4, branch in response to the fractional solution that is now produced in node 1 and produce affine functions g_5 and g_6 .

The function arising from node 5 coincides with the value function on the interval $(0, \frac{14}{3})$. The same tree is optimal for $\beta = 10$ and the optimal solution is obtained by re-optimizing the leaf nodes. ■

An important observation from Example 4 is that in warm-starting the computation using a given tree, the function arising from the procedure of Theorem 2 may change, although no further branching has been performed (this happened in the last iteration of the above example). This results from the re-optimization of the LP relaxation in each leaf node, which could yield a different solution to the continuous restriction. With respect to this new solution, it could be the case that variables whose values were previously integer become fractional (this can happen for integer variables whose values are not completely fixed by branching yet), requiring further branching. However, if we were to discard previously generated dual solutions in this case, important information would be lost, as we will see, and the function in one iteration might no longer be strong for right-hand sides from previous iterations. What is needed is a continuously refined dual function that remains strong for all previously arising right-hand side values and is made strong for a new right-hand side in a given iteration.

To maintain a single dual function that is strong for all previously occurring right-hand sides within a single tree, we must maintain a collection of all the linear functions that provide lower bounds for any given node, including all functions generated in previous iterations in either that node or any of its ancestors. Taking the maximum of all linear functions in this collection yields a convex lower approximation of the value function of the subproblem associated with that node (which is comprised of the original MILP plus some branching constraints). This function can be interpreted as a convex lower approximation of the value function of the LP relaxation of the given node (which is itself the best convex lower approximation of the value function of that subproblem). We can interpret the process of collecting these linear functions associated with each node as a process of building a convex approximation of its value function.

When branching occurs, the collection of linear functions of a given node must be inherited by *both* children. Branching can be interpreted as a process of identifying a region in which a convex lower approximation is not strong enough and then branching in order to divide the region into multiple subregions, each of which has its own convex bounding function. Through this process, we progressively improve the strength of the overall function. It should be clear that when doing the partitioning, we need to begin with the description of the original convex function in both of the child nodes and proceed from there. Naturally, in practice, many of the functions generated at higher levels of the tree will eventually be redundant and can be discarded.

To formalize this, let us consider the strongest bound that could be obtained by considering a given branch-and-bound tree if we are allowed to re-optimize the LP relaxations in each of the leaf nodes when evaluating the function with respect to a new right-hand side. Allowing the optimal solution to the LP relaxation to be changed without changing the tree yields a dual function stronger than the one described in Theorem 2 that is obtained by taking the minimum over the *full value functions* of the LP relaxations of each of the nodes in the tree. We define this strengthened function as

$$\underline{\phi}^*(\beta) = \min_{t \in T} \phi^t(\beta). \quad (4.11)$$

Going a step further, we note that if some of the branching bounds have served to fix the values of certain integer variables, we can explicitly indicate that these variables can be treated as constants

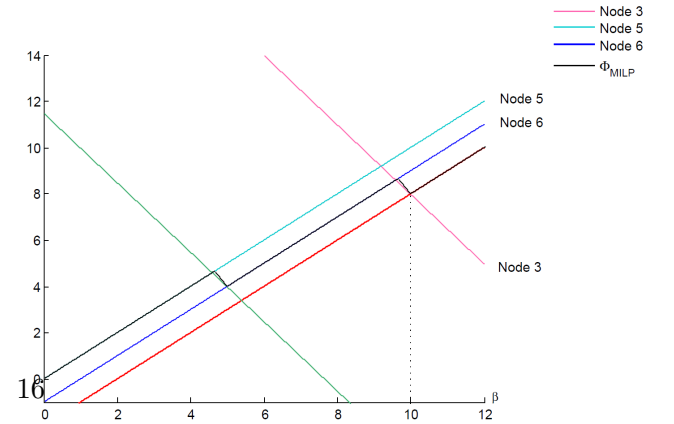
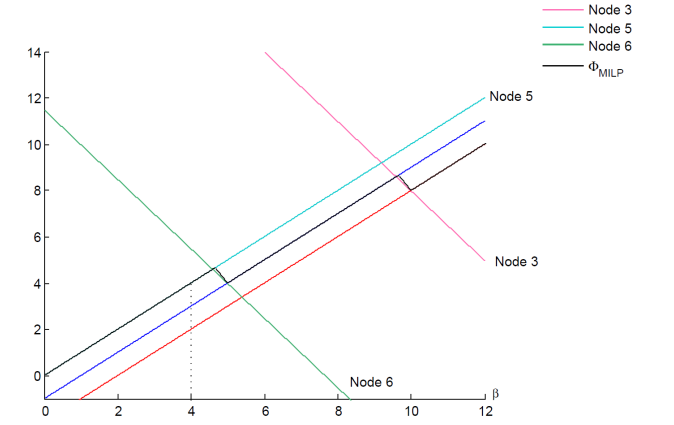
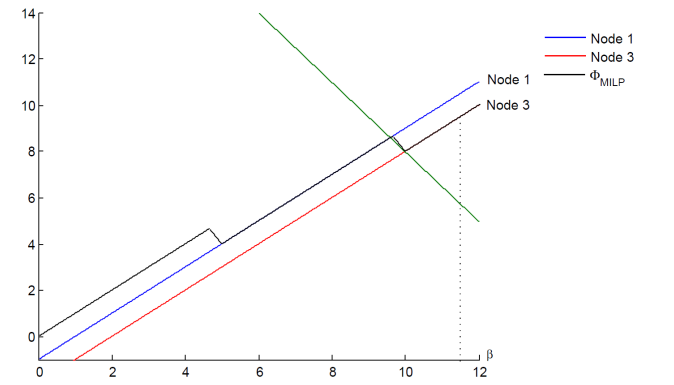
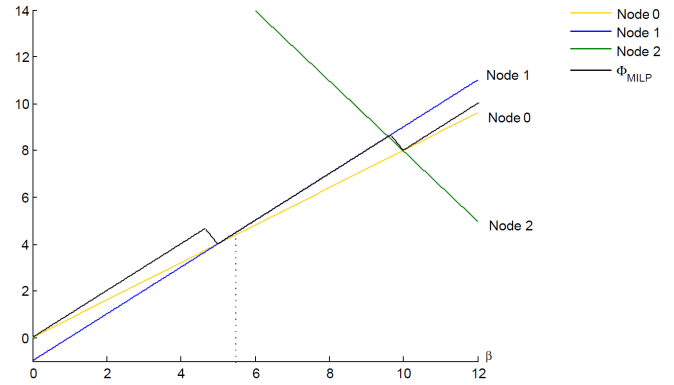
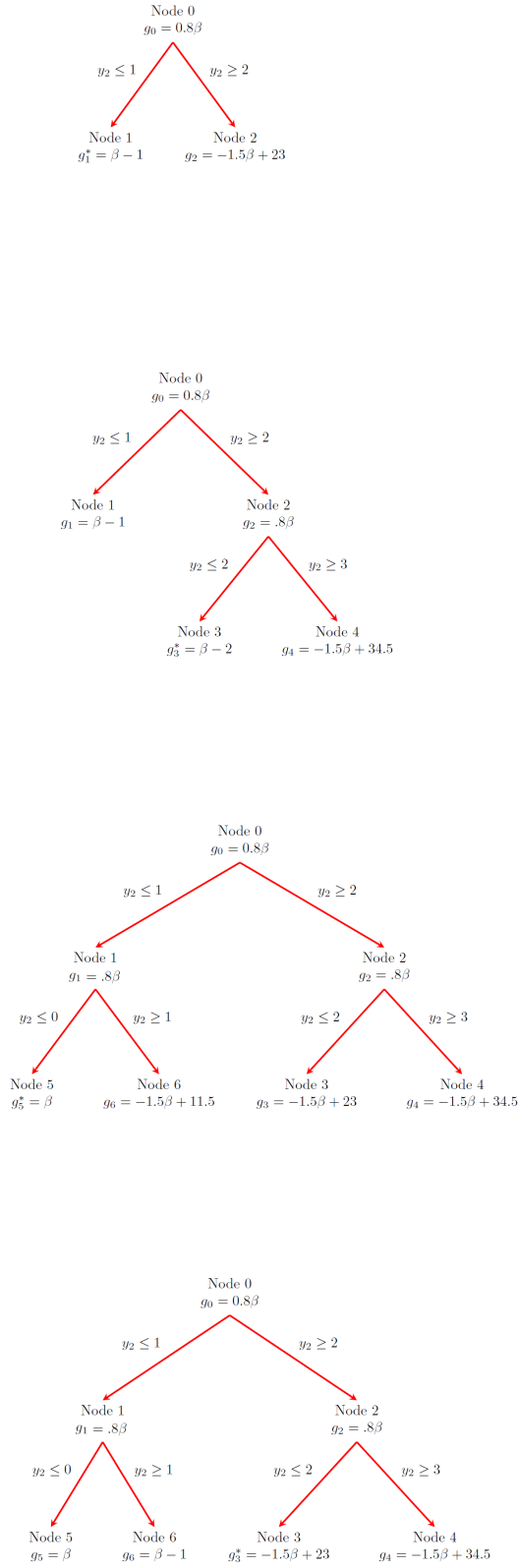


Figure 4: Strong dual functions from warm-starting branch-and-bound

to yield another equivalent definition.

$$\underline{\phi}^*(\beta) = \min_{t \in T} q_{I_t}^\top y_{I_t}^t + \phi_{N \setminus I_t}(\beta - W_{I_t} y_{I_t}^t), \quad (4.12)$$

where I_t is the set of indices of fixed variables, $y_{I_t}^t$ are the values of the corresponding variables in node t , and $\phi_{N \setminus I_t}$ is the value function of the linear program including only the unfixed variables.

With this strengthening, we obtain a function from a single tree that is the minimum over a collection of convex functions, just as the value function itself is. In fact, when the branching process is carried to its logical extreme, the above strengthened procedure will eventually yield the full value function. The fact that we can obtain the full value function from a single tree can be observed from the strong connection between the function (4.12) and the representation of the full value function from Theorem 1. If our branching decisions eventually lead to the fixing of variables in such a way that each member of \mathcal{S} is represented among the fixed portions of solutions in some set of leaf node, this would be enough to ensure that the tree would in fact represent the entire value function. This can be stated formally as the following result.

Theorem 3 *Under the assumption that $\{\beta \in \mathbb{R}^{m_2} \mid \phi_I(\beta) < \infty\}$ is finite, there exists a branch-and-bound tree with respect to which $\underline{\phi}^* = \phi$.*

Proof. To construct such a tree, we need to impose branching decisions that guarantee that $\mathcal{S} \subseteq \{y_{I_t} \mid t \in T, |I_t| = r_2\}$. Such a set of branching decisions can be easily constructed. ■

Returning to the second-stage problem in Example 3, we can see graphically that the members of the set \mathcal{S} from Theorem 1 that lie in the interval $(-10, 10)$ is

$$\{(0, 0, 5), (0, 0, 4), (0, 0, 3), (0, 0, 2), (0, 0, 1), (0, 0, 0), (0, 1, 0), (0, 2, 0)\}. \quad (4.13)$$

Therefore, over the domain $(-10, 10)$, we have

$$\begin{aligned} \phi(\beta) = \min \{ & 15 + \phi_C(\beta + 10), 12 + \phi_C(\beta + 8), 9 + \phi_C(\beta + 6), 6 + \phi_C(\beta + 4), \\ & 3 + \phi_C(\beta + 2), \phi_C(\beta), 4 + \phi_C(\beta - 5), 8 + \phi_C(\beta - 10) \}. \end{aligned} \quad (4.14)$$

To obtain this function from a given tree, we must ensure that each member of the above set produced by solutions of the LP relaxation of some leaf node in the tree *over the entire interval for which the piece coincides with the value function*. Figure 5 illustrates this principle by showing how to obtain the value function of the second-stage problem in Example 3 from a single branch-and-bound tree over the given interval. This is a minimal such tree—each leaf node corresponds to one affine piece of the value function. That is, the leaf nodes correspond exactly to the members of \mathcal{S} from Theorem 1. In general, the described procedure may result in a larger tree that contains extra affine functions in its nodes—these can be safely discarded. Figure 5 also shows the stability sets corresponding to each node.

5 The Algorithm

We now come to the formal statement of our generalization of Benders' method. At a high level, the approach consists of solving the value function reformulation (VF), generating relevant parts of

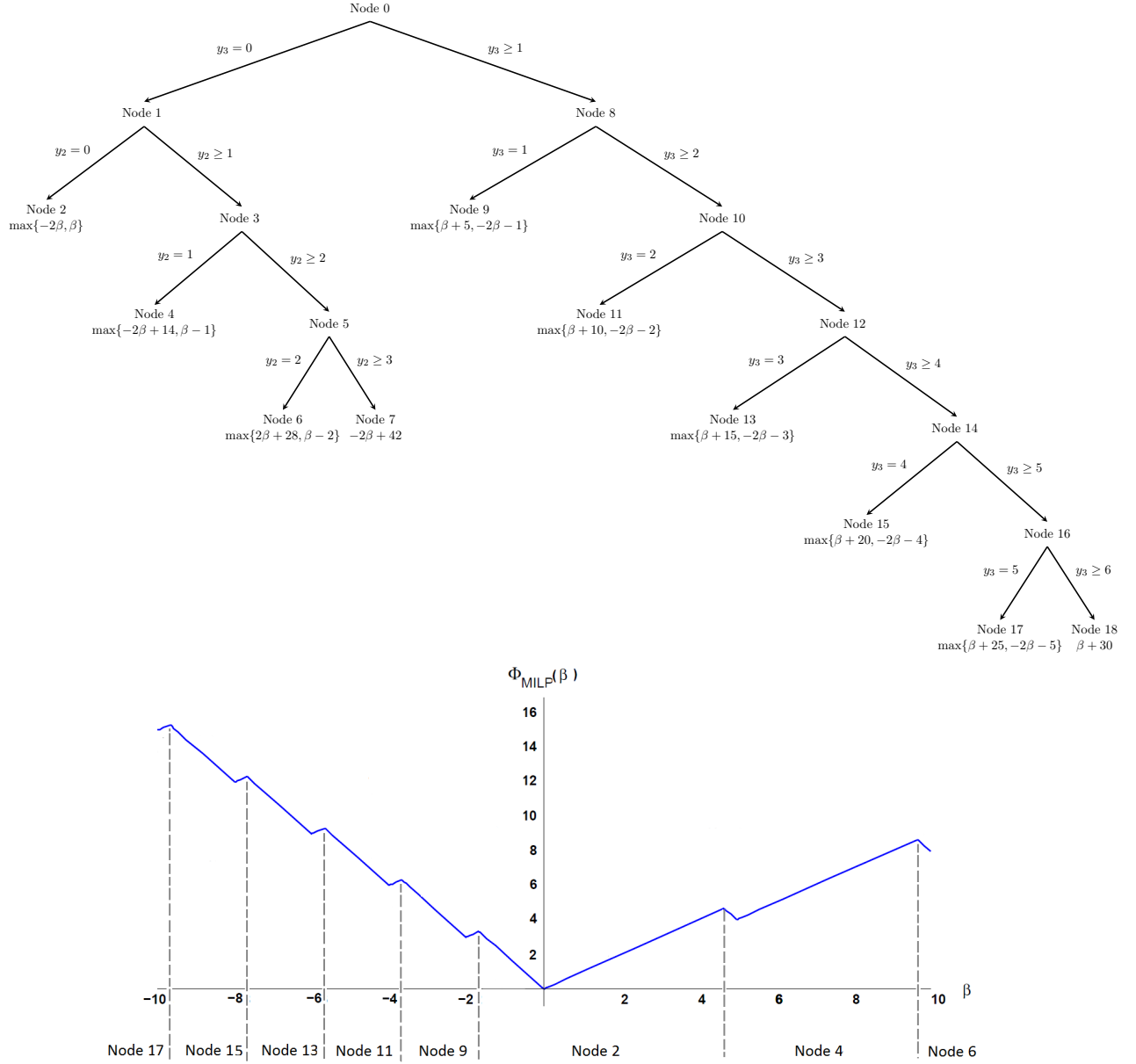


Figure 5: Branch-and-bound nodes and MILP value function correspondence.

the formulation dynamically, as in a traditional cutting plane method. Viewed in terms of the dual functions introduced in the previous section, the dynamic generation consists of further refinements of the branch-and-bound tree in which we solve instances of the second-stage and which we use to derive our current value function approximation.

The version of Benders' method we are proposing here differs from a classical implementation, such as the one first suggested by [Carøe and Tind \(1998\)](#) in this context, in that a straightforward interpretation of Benders' method would lead to generation of a new strong dual function in each iteration for each scenario, while we are simply refining the previously constructed dual function using the method described in the previous section. From a theoretical standpoint, these interpretations amount to the same thing. In terms of implementational details, however, the difference is quite significant. To be consistent with the traditional interpretation, we initially describe the method as generating a different dual function in each iteration before being more specific about the details of our implementation.

The first step in specifying the algorithm is to formulate the master problem. In a fashion similar to (3.6), we begin by first re-writing problem (SP) as

$$\min\{c^\top x + \theta \mid \theta \geq \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x), x \in \mathcal{P}_1\}. \quad (5.1)$$

As in the classical Benders' method, the idea of our generalization is to approximate the right-hand side of the first set of constraints in (5.1) by a set \mathcal{F}_ω of dual functions for each scenario and to iteratively strengthen the approximation yielded by these dual functions through the generation of additional such functions. To form the master problem, we therefore replace the value function with such an approximation to obtain

$$\min\{c^\top x + \theta \mid \theta \geq \sum_{\omega \in \Omega} p_\omega \max_{f \in \mathcal{F}_\omega} f(h_\omega - T_\omega x)\}, \quad (5.2)$$

where \mathcal{F}_ω represents the set of all dual functions associated with scenario ω that have been generated so far. In iteration k , a strong dual function f_ω^k is produced for each scenario with respect to a proposed first-stage solution $x^k \in \mathcal{P}_1$, the solution to the master problem in the previous iteration, by solving the dual problem

$$\max_f \{f(h_\omega - T_\omega x^k) \mid f(Wy) \leq q^\top y\}. \quad (5.3)$$

The collection of dual functions is then enlarged appropriately. With this approximation, the master problem after k iterations of the algorithm is

$$\begin{aligned} \min \quad & c^\top x + \theta \\ \text{s.t.} \quad & \theta \geq \sum_{\omega \in \Omega} p_\omega \max_{i=1, \dots, k} f_\omega^i(h_\omega - T_\omega x) \\ & x \in \mathcal{P}_1. \end{aligned} \quad (5.4)$$

In what follows, we denote the set of dual functions f_ω^i generated for the scenario ω in iterations $i = 1, \dots, k$ by \mathcal{F}_ω^k .

Generic Version of Benders' Algorithm

Step 0. Initialize

- a) Initialize the dual function lists $\mathcal{F}_\omega^0 := \emptyset$ for all $\omega \in \Omega$.
- b) Let $x^1 := \operatorname{argmin}\{c^\top x : x \in \mathcal{P}_1\}$, $\theta^1 := -\infty$ and $k := 1$.

Step 1. Update the lower approximation function

- a) For each $\omega \in \Omega$, solve the subproblem (RV) for the right-hand side $h_\omega - T_\omega x^k$ and construct the function f_ω^k dual to ϕ and strong at $h_\omega - T_\omega x^k$.
- b) Check whether the current approximation is exact for x^k . Stop if $\theta^k = \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x^k)$. $x^* := x^k$ is an optimal solution.

Step 2. Solve the master problem

- a) Update the dual functions list: $\mathcal{F}_\omega^k := \mathcal{F}_\omega^{k-1} \cup \{f_\omega^k\}$.
 - b) Solve (5.4) to obtain optimal solution (x^{k+1}, θ^{k+1}) to an optimal solution of it.
 - c) Set $k := k + 1$. Go to Step 1.
-

As described in the previous section, our implementation of this generic algorithm utilizes a single global dual function that is a slight variant of (4.12), rather than individual dual functions for each scenario. For efficiency, we do not explicitly derive a complete description of the value function of each LP relaxation, but generate pieces of it dynamically as part of the overall algorithm. This dual function consists of the minimum of a collection of piecewise polyhedral convex functions (one for each leaf node in the branch-and-bound tree). This makes the master problem a nonlinear optimization problem involving this single piecewise polyhedral function for which we do not have explicit descriptions of the polyhedral regions over which it is affine. Nevertheless, it is possible to reformulate (5.4) as a standard MILP similar to (VF) by introducing auxiliary variables and constraints.

Let T be the set of leaf nodes of the current tree. Recall that corresponding to each leaf node, we have a set of affine dual functions that have been obtained in the leaf node and its ancestors as we refined the tree in solving previous subproblems. Let $\mathcal{I}(t)$ denote the set of indices of these affine functions at a leaf node $t \in T$. The dual function obtained from the current tree is

$$\underline{\phi}(\beta) = \min_{t \in T} \max_{i \in \mathcal{I}(t)} \beta^\top \pi^i + \alpha^i. \quad (\text{DF})$$

As explained earlier, π and α can be obtained from (4.6) and (4.8). We can then write (5.4) as the following MILP.

$$\begin{aligned}
\Gamma^k &= \min c^\top x + \sum_{\omega \in \Omega} p_\omega \underline{\phi}_\omega \\
\text{s.t. } \underline{\phi}_\omega &\leq q_{t,\omega} & \forall t \in T, \omega \in \Omega \\
\underline{\phi}_\omega &\geq q_{t,\omega} - M_\omega(1 - u_{t,\omega}) & \forall t \in T, \omega \in \Omega \\
q_{t,\omega} &\geq (h_\omega - T_\omega x)^\top \pi^i + \alpha^i & \forall i \in \mathcal{I}(t), t \in T, \omega \in \Omega \\
\sum_{t \in T} u_{t,\omega} &= 1 & \forall \omega \in \Omega \\
u_{t,\omega} &\in \mathbb{B}, & \forall t \in T, \omega \in \Omega \\
x &\in X.
\end{aligned} \tag{MP}$$

In (MP), $q_{t,\omega}$ represents the piecewise convex function obtained at a leaf node of the tree. That is, the fourth constraint ensures that $q_{t,\omega} = \max_{i \in \mathcal{I}(t)} \beta^\top \pi^i + \alpha^i$. $\underline{\phi}_\omega$ is the approximation of recourse for scenario ω . The second constraint guarantees that $\underline{\phi}_\omega$ is less than all the convex approximations obtained at the leaf nodes, while the third and fifth constraints guarantee that one of them holds at equality and $\underline{\phi}_\omega$ achieves the minimum of the leaf nodes. In the third constraint, M_ω is an appropriately large positive number. For instance, $M_\omega \geq \max_{t \in T} |q_{t,\omega}|$. Before formally proving correctness and finiteness, we illustrate the algorithm with an example where we show the details of this approach.

Example 5. We consider a modified version of Example 3 for which $\mathcal{P}_1 = \{x_1, x_2 \in \mathbb{B} \mid x_1 + x_2 \leq 1\}$. The sequence of subproblems solved here is the same as the sequence from Example 4, but we now show in Figure 6 the full details of the single tree approach, including retention of all linear pieces to ensure a dual that is strong for all evaluated right-hand sides. The starting point $(x_1^1, x_2^1) = (0, 1)$ in Step 0 is obtained through solving

$$\Gamma^1 = \min\{-3x_1 - 3.8x_2 \mid x_1 + x_2 \leq 1, x_1, x_2 \in \mathbb{B}\}.$$

Let $\theta^1 = -\infty$ and $k = 1$.

Iteration 1. In Step 1, we solve the second-stage problem for $6 - 0.5$ and $12 - 0.5$ with branch and bound to obtain the dual function

$$\underline{\phi}^1(\beta) = \min\{\max\{\beta - 1, 0.8\beta\}, \max\{\beta - 2, 0.8\beta, -1.5\beta + 23\}, \max\{-1.5\beta + 34.5, 0.8\beta\}\}.$$

Note that we retain the linear piece from solving the root node in each of the children, which yields a stronger dual function overall, though this piece is not required for strength at either of the two right-hand sides in the first iteration. The updated master problem is

$$\Gamma^2 = \min\{-3x_1 - 3.8x_2 + 0.5\underline{\phi}^1(6 - 2x_1 - 0.5x_2) + 0.5\underline{\phi}^1(12 - 2x_1 - 0.5x_2) \mid x_1 + x_2 \leq 1, x_1, x_2 \in \mathbb{B}\}.$$

This would normally be formulated as in (MP) and solved as a generic MILP in practice, but for clarity and compactness, we avoid this step here. The solution is $\Gamma^2 = 2.6$ with $\theta^1 = 5.6$ and $(x_1^2, x_2^2) = (1, 0)$. Let $k = 2$.

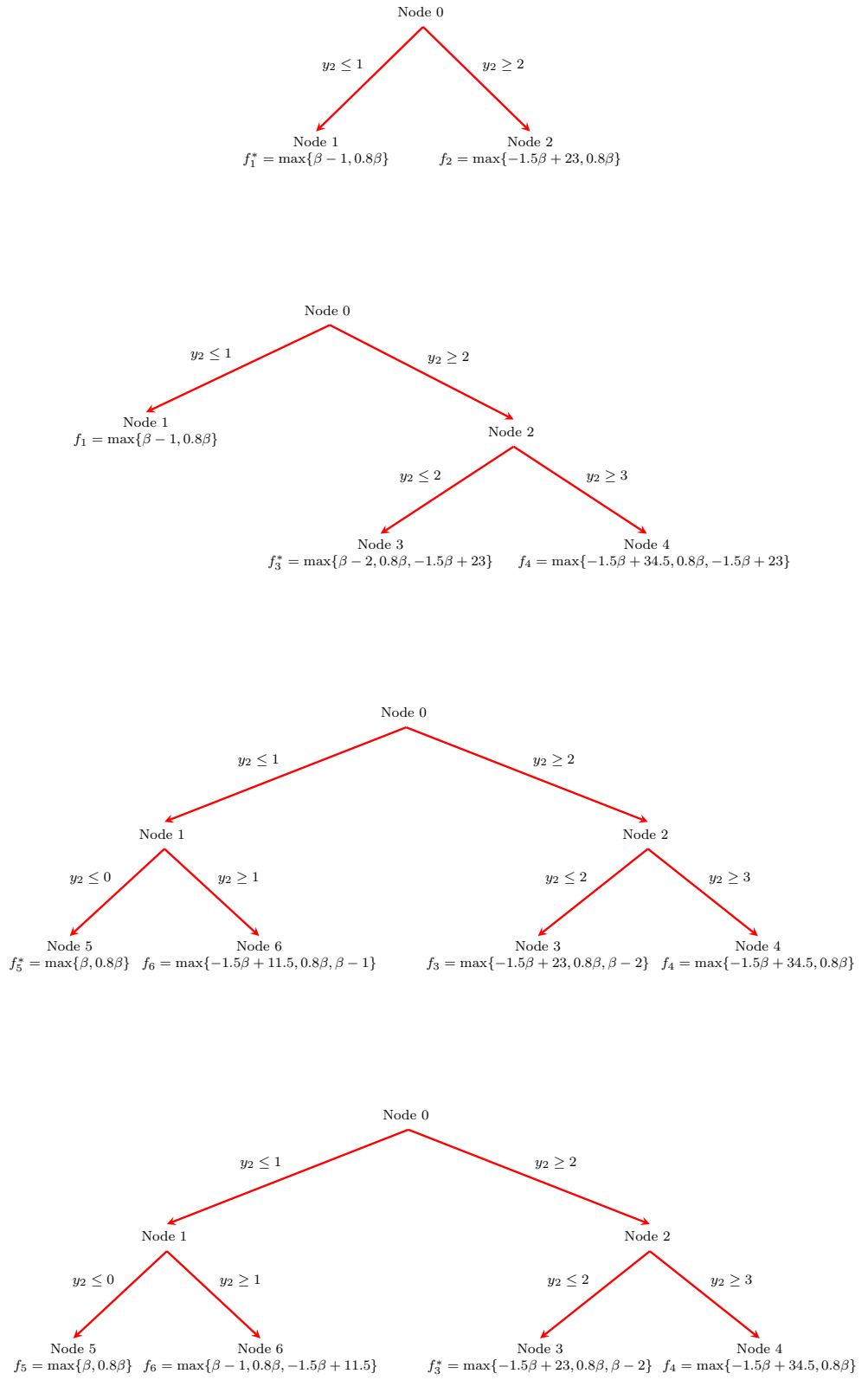


Figure 6: Strengthened dual functions from warm-starting

Iteration 2. We solve the subproblem with the right-hand sides 4 (corresponding to the first scenario) and 10 (corresponding to the second scenario) to obtain our second dual function

$$\begin{aligned}\underline{\phi}^2(\beta) = \min\{ & \max\{\beta, 0.8\beta\}, \\ & \max\{-1.5\beta + 11.5, 0.8\beta, \beta - 1\}, \\ & \max\{-1.5\beta + 23, 0.8\beta, \beta - 2\}, \\ & \max\{-1.5\beta + 34.5, 0.8\beta\}\} \end{aligned} \quad (5.5)$$

Since $5.6 < 0.5(4 + 8)$, we continue. In Step 2, we solve the updated master problem

$$\Gamma^3 = \min\{-3x_1 - 3.8x_2 + 0.5\underline{\phi}^2(6 - 2x_1 - 0.5x_2) + 0.5\underline{\phi}^2(12 - 2x_1 - 0.5x_2) \mid x_1 + x_2 \leq 1, x_1, x_2 \in \mathbb{B}\}.$$

to obtain $\Gamma^3 = 3$ with $\theta^3 = 6$ and $(x_1^3, x_2^3) = (1, 0)$. We let $k = 3$.

Iteration 3. The solution x^3 is identical to x^2 . Since $\theta^3 = 0.5(\underline{\phi}^2(4) + \underline{\phi}^2(10))$, we stop. $(x_1^*, x_2^*) = (1, 0)$ is an optimal solution.

■

Having now demonstrated the algorithm, we proceed to formally state the following result.

Theorem 4 *Consider the generalization of Benders' algorithm in which*

- *dual functions of the form (DF) are generated within a single branch-and-bound tree and*
- *a master problem of the form (MP) is utilized.*

This algorithm terminates with the correct global minimum in finitely many steps.

Proof. We first show that if the termination check is satisfied, the algorithm terminates with a correct optimal solution. Assume the algorithm terminates in iteration k with the dual function obtained from tree T^{k-1} and let x^* be the optimal solution to the problem. Then, we have

$$\begin{aligned} c^\top x^k + \sum_{\omega \in \Omega} p_\omega \min_{t \in T^{k-1}} \max_{i \in \mathcal{I}(t)} \{ (h_\omega - T_\omega x^k)^\top \pi^i + \alpha^i \} &= c^\top x^k + \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x^k) \\ &\leq c^\top x^* + \min_{t \in T^{k-1}} \max_{i \in \mathcal{I}(t)} \{ (h_\omega - T_\omega x^*)^\top \pi^i + \alpha^i \} \\ &\leq c^\top x^* + \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x^*) \\ &\leq c^\top x^k + \sum_{\omega \in \Omega} p_\omega \phi(h_\omega - T_\omega x^k). \end{aligned}$$

where the first line is from the termination condition, the second line follows from the optimality of x^k for the master problem in iteration k , the third line follows since (DF) is a dual function for the value function of the recourse and the last line holds since x^* is the optimal solution to the problem.

To show the finiteness of the algorithm, note that if any first-stage solution of the master problem repeats in a future iteration, the termination criterion is satisfied and we have an optimal solution. This holds because the dual function obtained from an optimal tree is strong with respect

to all previous right-hand sides. Therefore, if the same first-stage solution repeats, the dual function will already be strong in all scenarios and the termination check will then be satisfied. Furthermore, observe that in each iteration, we must either generate at least one new linear dual function within the branch-and-bound tree at some node or branch on some node. From Assumption A2, the set \mathcal{B} defined in (3.5) is finite. There are a finite number of branching operations that can be done before achieving a complete description of this set. Assumption A1 also implies that there is a finite number of extreme points and rays of the dual polyhedron \mathcal{D} of the recourse problem. Together, we have a finite number of operations that can occur during this process and a finite number of dual functions that can arise. In a finite number of steps, we therefore generate the full formulation (VF), at which point the tree must contain a full description of the value function over B and the process must terminate. ■

6 Conclusions

We propose a reformulation of the general two-stage stochastic mixed integer linear optimization problem based on a discrete representation of the value function and show how that leads to a generalization of Benders’ algorithm. Our implementation of the algorithm utilizes dual functions obtained as by-products of the branch-and-bound procedure for solving the second-stage problem. Such branch-and-bound dual functions, if derived for each scenario in each iteration, yield a convergent version of Benders’ method. Solving the subproblems from scratch and incorporating the large number of dual functions that would arise into the master problem, however, seems unlikely to result in a scalable algorithm. To address the challenge of this very difficult problem class, we propose to improve the basic method with the addition of a warm-starting strategy in which all subproblems are solved within a single branch-and-bound tree and in which we use a single continuously refined lower approximation of the value function within the master problem. By taking a global view of the construction of dual functions, we are able to derive stronger dual functions and avoid redundancy in the formulation of the master problem. As a corollary to our approach, we have further shown that there exists a single branch-and-bound tree from which we can derive the full value function of the second-stage problem, though it is unnecessary to do so in practice.

Our approach is specifically designed to yield an algorithm that can be implemented in practice and we have such an implementation. However, it remains to be seen how scalable this method will be in practice. Despite our focus on practical implementation, what we have presented here is a primarily theoretical treatment and we believe the results are of interest, even at a theoretical level. Many questions remain to be answered about how to manage the execution of this method in a real implementation. It is clear that a key aspect will be not just how to dynamically add relevant parts to the formulation, but how to delete parts that have become irrelevant. Many of the issues are similar to those that arise in the management of state-of-the-art branch-and-cut implementations. A final question is whether it would be fruitful to extend our method of constructing dual functions to allow for cut generation within the branch-and-bound search. It is not at all clear how advantageous this would be, given that in the stochastic context, some amount of enumeration is necessary to fully describe the value function. Further extensions of this work along with a computational study will be reported in a subsequent paper.

References

- Ahmed, S., Tawarmalani, M., and Sahinidis, N. (2004). A finite branch-and-bound algorithm for two-stage stochastic integer programs. *Mathematical Programming*, 100(2):355–377.
- Avis, D. and Fukuda, K. (1992). A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(1):295–313.
- Bank, B., Guddat, J., Klatte, D., Kummer, B., and Tammer, K. (1983). *Non-linear parametric optimization*. Birkhäuser verlag.
- Blair, C. and Jeroslow, R. (1977). The value function of a mixed integer program: I. *Discrete Mathematics*, 19(2):121–138.
- Blair, C. and Jeroslow, R. (1982). The value function of an integer program. *Mathematical Programming*, 23(1):237–273.
- Carøe, C. and Schultz, R. (1998). Dual decomposition in stochastic integer programming. *Operations Research Letters*, 24(1):37–46.
- Carøe, C. and Tind, J. (1997). A cutting-plane approach to mixed 0-1 stochastic integer programs. *European Journal of Operational Research*, 101(2):306–316.
- Carøe, C. and Tind, J. (1998). L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83(1):451–464.
- Gade, D., Küçükyavuz, S., and Sen, S. (2012). Decomposition algorithms with parametric Gomory cuts for two-stage stochastic integer programs. *Mathematical Programming*, pages 1–26.
- Güzelsoy, M. (2009). *Dual methods in mixed integer linear programming*. PhD thesis, Lehigh University.
- Güzelsoy, M. and Ralphs, T. (2007). Duality for mixed-integer linear programs. *Internat. J. Oper. Res*, 4(3):118–137.
- Hassanzadeh, A., Ralphs, T. K., and Güzelsoy, M. (2014). On the value function of a mixed integer linear optimization problem and an algorithm for construction. Technical report, COR@L Laboratory, Lehigh University.
- Kong, N., Schaefer, A., and Hunsaker, B. (2006). Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Mathematical Programming*, 108(2):275–296.
- Laporte, G. and Louveaux, F. (1993). The e for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142.
- Ntaimo, L. (2010). Disjunctive decomposition for two-stage stochastic mixed-binary programs with random recourse. *Operations research*, 58(1):229–243.
- Ralphs, T. K. and Güzelsoy, M. (2005). The SYMPHONY callable library for mixed-integer linear programming. In *Proceedings of the Ninth INFORMS Computing Society Conference*, pages 61–76.

- Ralphs, T. K. and Güzelsoy, M. (2006). Duality and warm starting in integer programming. In *The Proceedings of the 2006 NSF Design, Service, and Manufacturing Grantees and Research Conference*.
- Schultz, R., Stougie, L., and Van Der Vlerk, M. (1998). Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis. *Mathematical Programming*, 83(1):229–252.
- Sen, S. and Higle, J. (2005). The C3 theorem and a D2 algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104(1):1–20.
- Sen, S. and Sherali, H. (2006). Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223.
- Sherali, H. and Fraticelli, B. (2002). A modification of Benders’ decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22(1):319–342.
- Sherali, H. and Zhu, X. (2006). On solving discrete two-stage stochastic programs having mixed-integer first-and second-stage variables. *Mathematical Programming*, 108(2):597–616.
- Sherali, H. D. and Smith, J. C. (2009). Two-stage stochastic hierarchical multiple risk problems: models and algorithms. *Mathematical programming*, 120(2):403–427.
- Trapp, A. C., Prokopyev, O. A., and Schaefer, A. J. (2013). On a level-set characterization of the value function of an integer program and its application to stochastic programming. *Operations Research*, 61(2):498–511.
- Van Slyke, R. and Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, pages 638–663.
- Wolsey, L. (1981). Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming*, 20(1):173–195.
- Yuan, Y. and Sen, S. (2009). Enhanced cut generation methods for decomposition-based branch and cut for two-stage stochastic mixed-integer programs. *INFORMS Journal on Computing*, 21(3):480–487.