Industrial and
Systems Engineering

A Framework for Generalized Benders' Decomposition and Its Application to Multilevel Optimization

SURESH BOLUSANI AND TED K. RALPHS

Department of Industrial and System Engineering, Lehigh University, Bethlehem, PA

COR@L Technical Report 20T-004

LEHIGH UNIVERSITY

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

A Framework for Generalized Benders' Decomposition and Its Application to Multilevel Optimization

Suresh Bolusani*[1] and Ted K. Ralphs†[1]

[1]Department of Industrial and System Engineering, Lehigh University, Bethlehem, PA

Abstract

We describe an algorithmic framework generalizing the well-known framework originally introduced by Benders. We apply this framework to several classes of optimization problems that fall under the broad umbrella of multilevel/multistage mixed integer linear optimization problems. The development of the abstract framework and its application to this broad class of problems provides new insights and new ways of interpreting the core ideas, especially those related to duality and the value function of an optimization problem.

1 Introduction

This paper describes an algorithmic framework that extends the well-known framework of Benders [1962] and illustrates the principles involved by applying them to the solution of several well-known classes of optimization problems. These classes of problems are all contained under the broad umbrella of what we informally refer to as *multilevel/multistage mixed integer linear optimization problems* (MMILPs). MMILPs comprise a broad class of optimization problems in which multiple decision makers (DMs), with possibly competing objectives, make decisions in sequence over time. Each DM's decision impacts the options available to other DMs at other (typically later) stages. In economics, these problems fall under the general umbrella of game theory. We do not formally define the broad class comprising MMILPs here, but rather describe some specific subclasses contained within it. Readers wishing to have a more complete overview of MMILPs should refer to Bolusani et al. [2020].

MMILPs have a natural recursive structure, which lends itself to solution using Benders' original approach. An l-stage problem is most naturally defined recursively in terms of an l-1-stage problem. This recursive structure mirrors that of the *polynomial time hierarchy* (PTH), a recursively defined family of complexity classes into which MMILPs can naturally be categorized. The lowest level of the PTH is the well-known class P of problems solvable in time polynomial in the size of the input, and the l^{th} level (denoted Σ_P^l) is comprised of problems solvable in polynomial time given an oracle

*bsuresh@lehigh.edu

†ted@lehigh.edu

for problems in the $(l\text{-}1)^{\text{st}}$ level. MMILPs with l levels are prototypical complete problems for the l^{th} level of the PTH.

Although the application of Benders' original algorithm was to standard mathematical optimizations problems with an underlying structure that suggested a natural division of the variables into two groups, it can be similarly applied not only to two-stage MMILPs, but by extension, to l-stage problems in which there is a natural division of the variables into l groups. Benders' framework can be applied first and foremost in simply reformulating such problems as standard mathematical optimization problems, with the reformulation suggesting an associated algorithmic approach that can be applied recursively, essentially decomposing the problem by stage. In such an approach, the subproblem arising when solving an l-stage problem is a (lexicographic) optimization problem with $l - 1$ stages.

This work is a natural successor to that of Hassanzadeh and Ralphs [2014a], who utilized a generalization of Benders' approach that can be seen as a special case of the one described here, to develop an algorithm to solve two-stage stochastic mixed integer linear optimization problems with recourse (2SSMILPs). In order to emphasize the connection to stochastic optimization and also because of its broader connotation, we use the term "stage" throughout this paper in describing the decision epochs of an MMILP, rather than the more standard "level." Thus, we use the term "multistage optimization problems" informally to refer to MMILPs.

The remainder of the paper is organized as follows. In Section 2, we discuss the principles underlying our generalized Benders' framework at a high level in the context of a general optimization problem, including concepts of bounding functions and general duality. In Section 3, we illustrate these principles concretely with two examples, summarizing existing algorithms for the case in which there are two stages and the objectives functions are the same in both stages. Section 4 goes into more detail in describing an algorithm for general *two-stage/bilevel mixed integer linear optimization problems* (MIBLPs), the special case of MMILPs in which there are only two stages. Finally, in Section 5, we discuss further extensions to general multistage problems before concluding in Section 6.

We emphasize that while we have implemented the algorithms described here, the implementation is naive and only meant as a proof of concept. This paper is not aimed at discussing efficiency or comparing the algorithms described herein to alternatives. For the algorithmic framework described here to be brought to fruition with efficient implementations, substantial additional development is required.

2 Benders' Principle

In this section, we introduce the basic principles of the framework. We first describe it in a very general context and then focus on the special case in which the objective and constraint functions are *additively separable* (defined formally below). The idea of such a generalization of Benders' original algorithm is not new. As far back as the 1970s, Geoffrion [1972] had already proposed a similar idea. Its application to MMILPs, however, provides new insights and new ways of interpreting the core ideas.

We first consider the following general optimization problem in which the variables are partitioned into two sets. We refer to these sets as the *first-stage variables*, denoted by $x \in \mathbb{R}^{n_1}$, and the *second-stage variables*, denoted by $y \in \mathbb{R}^{n_2}$. The problem is then

$$\min_{x\in X,\ y\in Y} \{f(x,y) \mid F(x,y) \geq 0\}, \tag{GP}$$

where $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ is the objective function and $F : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^m$ is the constraint function, with $X \subseteq \mathbb{R}^{n_1}_+$ and $Y \subseteq \mathbb{R}^{n_2}_+$ denoting the additional disjunctive constraints on the values of the variables. Typically, we have $X = \mathbb{Z}^{r_1}_+ \times \mathbb{R}^{n_1-r_1}_+$ and $Y = \mathbb{Z}^{r_2}_+ \times \mathbb{R}^{n_2-r_2}_+$, and we therefore consider that form for the remainder of the paper. By convention, we take the optimal objective value to be ∞ if the feasible region

$$\mathcal{F} = \{(x,y) \in X \times Y \mid F(x,y) \geq 0\}$$

is empty and $-\infty$ if the problem (GP) is unbounded. We assume that in all other cases, the problem has a finite minimum that can be attained.

The most succinct way of describing Benders' approach is that it projects (GP) into the space of the first-stage variables. In this way, we obtain a reformulation involving only the first-stage variables. The projection operation is natural in applications where the optimal values of the first-stage variables are our primary concern, while the second-stage variables are present only to model the later-stage effects of the first-stage decisions.

As we show shortly, the reformulation process necessarily introduces complex functions of the first-stage variables, which model the effects mentioned above. Algorithms for solving these reformulations thus begin by bounding these complex functions from below via so-called *lower-bounding functions* to obtain a relaxation of the original problem, called the *master problem*, which is solved to yield a first-stage solution and a lower bound. The associated *subproblem*, on the other hand, is to solve the second-stage problem with this fixed first-stage solution to yield an upper bound, as well as a bounding function that is fed back into the master problem to strengthen the relaxation. This process is iterated until the upper and lower bounds are equal.

2.1 General Optimization Problems

In this section, we provide a formal description of the framework, assuming general objective and constraint functions, highlighting the basic algorithmic structure.

2.1.1 Projection and the Subproblem

The simple yet fundamental idea is to note that (GP) can be equivalently formulated as follows.

$$\min_{x\in X} \left\{ \min_{y\in Y} \{f(x,y) \mid F(x,y) \geq 0\} \right\} \tag{GP-Decomp}$$

Here, we have decomposed the optimization into two stages. By replacing the second-stage optimization problem with a function, we obtain the reformulation

$$\min_{x\in X} \phi_{SS}(x), \tag{GP-Proj}$$

where

$$\phi_{SS}(x) = \min_{y \in Y} \{f(x,y) \mid F(x,y) \geq 0\} \quad \forall x \in \mathbb{R}^{n_1}. \tag{GP-SP}$$

In this new formulation, ϕ_{SS} is a function that returns the optimal value of the second-stage problem as a function of the first-stage variables. By convention, $\phi_{SS}(x) = \infty$ if $x \notin \text{proj}_x(\mathcal{F})$, where

$$\text{proj}_x(\mathcal{F}) = \{x \in X \mid F(x,y) \geq 0 \text{ for some } y \in Y\} \tag{GP-FR-Proj}$$

is the projection of the feasible region of (GP) into the space of the first-stage variables, and $\phi_{SS} = -\infty$ if the optimization problem (GP-SP) is unbounded.

Although the formulation (GP-Proj) does not explicitly involve $\text{proj}_x(\mathcal{F})$, the fact that $\phi_{SS}(x) = \infty$ for $x \notin \text{proj}_x(\mathcal{F})$ implicitly disallows such first-stage solutions from being selected as an optimum (assuming $\mathcal{F}$ is non-empty). Hence, (GP-Proj) can be viewed simply as the projection of (GP) into the space of the first-stage variables.

The evaluation of ϕ_{SS} for particular first-stage solutions is what we referred to earlier as the *subproblem*. Although this evaluation apparently involves only the determination of the optimal solution value, in practice, we also try to extract information that can help in forming the lower-bounding function that is a crucial component of the master problem, described next.

2.1.2 Bounding Functions and the Master Problem

In principle, the optimal value of (GP), as well as an optimal first-stage solution, can be obtained by solving (GP-Proj). However, we usually do not have a closed-form description of ϕ_{SS} and even when such closed form exists in theory, its description is typically of exponential size and would thus be impractical. We therefore replace ϕ_{SS} in (GP-Proj) with a lower-bounding function $\underline{\phi}_{SS}$ (defined below) to obtain a relaxation we earlier referred to as the *master problem*. Although it should be intuitively clear what is meant by a "lower-bounding function," we provide a formal definition here.

Definition 1 (Lower-Bounding Function). *A function $\underline{\phi} : \mathbb{R}^{n_1} \to \mathbb{R} \cup \{\pm\infty\}$ is said to be a lower-bounding function of the function ϕ if*

$$\underline{\phi}(x) \leq \phi(x) \quad \forall x \in \mathbb{R}^{n_1}.$$

It is called strong *at $\hat{x} \in X$ if*

$$\underline{\phi}(\hat{x}) = \phi(\hat{x}).$$

Given a lower-bounding function, the master problem is then

$$\min_{x \in X} \underline{\phi}_{SS}(x). \tag{GP-MP-LB}$$

Naturally, for any relaxation-based method to be practical, solving the relaxation (in this case, (GP-MP-LB)) should be easier than solving the original problem (in this case, (GP-Proj)). The difficulty of solving (GP-MP-LB), however, is directly related to the structure of the function $\underline{\phi}_{SS}$ itself. In the cases discussed later, this function is piecewise linear and the master problem can be formulated as a mixed integer linear program.

2.1.3 Overall Algorithm

The overall method is to iteratively improve the master problem formulation by strengthening $\underline{\phi}_{SS}$. In iteration k, candidate solution x^k is produced by solving the current master problem and $\phi_{SS}(x^k)$ is evaluated, producing a primal-dual proof of optimality in the form of a lower-bounding function $\underline{\phi}^k_{SS}$ that is strong at x^k. We describe later in Section 2.2.2 what form this primal-dual proof may take in specific cases.

To ensure convergence of the method, a global approximation $\underline{\phi}_{SS}$ must be maintained that is strong not only for x^k, but for all x^i, $i \leq k$. This can be most easily accomplished by taking the maximum of the bounding functions generated at each iteration. That is, after iteration k,

$$\underline{\phi}_{SS}(x) = \max_{1\leq i\leq k} \underline{\phi}^i_{SS}(x), \tag{LBF}$$

where $\underline{\phi}^i_{SS}(x)$ is the lower-bounding function obtained in iteration $i \leq k$ of the algorithm. In such cases, the master problem (GP-MP-LB) is usually reformulated using a standard trick to eliminate the maximum operator by introducing an auxiliary variable z to obtain

$$\begin{aligned} &\min_{x\in X} z \\ &\text{s.t. } z \geq \underline{\phi}^i_{SS}(x) \quad 1 \leq i \leq k. \end{aligned} \tag{GP-MP}$$

The formulations (GP-MP) and (GP-MP-LB) are equivalent in this case, since z must be equal to the maximum of the individual bounding functions at optimality.

The constraints $z \geq \underline{\phi}^i_{SS}(x)$ for $1 \leq i \leq k$ in (GP-MP) are typically called *optimality constraints* in Benders' standard approach. Depending on how the master problem is reformulated, it may also sometimes be necessary to explicitly add separate constraints referred to as *feasibility constraints* whenever $x^k \notin \text{proj}_x(\mathcal{F})$. These constraints are unnecessary in principle, though, because we have $\underline{\phi}^k_{SS}(x^k) = \infty$ in such a case ("∞" would be replaced by a big-M for practical computations), and thus, a valid lower-bounding function would still suffice to ensure that such a solution would not be produced.

The algorithm continues, alternating between solution of a master problem, which produces a lower bound, and a subproblem, which produces an upper bound, until termination, which occurs when upper and lower bounds are equal. The overall approach is outlined in Figure 1. This general framework leaves many steps unspecified and raises many questions regarding implementation in specific cases. These questions will be answered in detail for the several cases we cover in Sections 3 and 4.

A question that has a more general answer, however, concerns convergence of the algorithm. It is possible in general that the algorithm will not converge at all or that it will converge to a local optimum instead of a global optimum [Sahinidis and Grossmann, 1991]. However, Hooker and Ottosson [2003] proved that the algorithm converges in a finite number of steps under two conditions that are satisfied in many important cases. Following the framework just described, one of these conditions is automatically satisfied, as indicated in the following theorem.

Theorem 1. *[Hooker and Ottosson, 2003] If the function $\underline{\phi}_{SS}$ is defined as in (LBF) and $\underline{\phi}^i_{SS}$ is strong at x^i in each iteration i, then $\underline{\phi}_{SS}$ remains a valid lower-bounding function that is strong at*

Generalized Benders' Framework for Solving (GP)

Step 0. Initialize $k \leftarrow 1$, $\mathrm{UB}^0 = \infty$, $\mathrm{LB}^0 = -\infty$, $\underline{\phi}^0_{SS}(x) = -\infty$ for all $x \in \mathbb{R}^{n_1}$.

Step 1. Solve the master problem (lower bound)

a) Construct the lower-bounding function $\underline{\phi}_{SS}(x) = \max_{0 \le i \le k-1} \underline{\phi}^i_{SS}(x)$ and formulate the master problem (GP-MP).

b) Solve (GP-MP) to obtain an optimal solution $(x^k,\ z^k)$. Set $\mathrm{LB}^k \leftarrow z^k$.

Step 2. Solve the subproblem (upper bound)

a) Solve (GP-SP) for the given x^k to obtain an optimal solution y^k and strong lower-bounding function $\underline{\phi}^k_{SS}$ such that $\underline{\phi}^k_{SS}(x^k) = \phi_{SS}(x^k)$. Set $\mathrm{UB}^k \leftarrow \phi_{SS}(x^k)$.

b) Termination check: $\mathrm{UB}^k = \mathrm{LB}^k$?

1. If yes, STOP. (x^k, y^k) is an optimal solution to (GP).
2. If no, set $k \leftarrow k+1$ and go to Step 1.

Figure 1: Outline of the generalized Benders' decomposition algorithmic framework

all previous iterates and the method converges to the optimal value in a finite number of iterations under the additional assumption that $|\operatorname{proj}_x(\mathcal{F})| < \infty$.

The proof of this result is rather straightforward. In fact, a slightly more general result also holds, since the function need not be constructed in this particular way, as long as we can ensure that the overall lower-bounding function is strong at all the previous iterates. In practical implementations, taking the maximum of previous bounding functions is a natural approach and we consider the same here.

2.2 Additively Separable Optimization Problems

We now move to the more specific setting that is central to the application of Benders' method to MMILPs, that in which the constraint and objective functions are additively separable.

Definition 2 (Additively Separable Function). *A function* $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$ *is* additively separable *if* $\exists\ g : \mathbb{R}^{n_1} \to \mathbb{R}$ *and* $h : \mathbb{R}^{n_2} \to \mathbb{R}$ *such that* $f(x, y) = g(x) + h(y)$ *for all* $(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$.

When the functions f and F are additively separable, such separability allows us to reformulate these problems in ways that enhance intuition and also ease implementation. As such, let $g : \mathbb{R}^{n_1} \to \mathbb{R}$, $h : \mathbb{R}^{n_2} \to \mathbb{R}$, $G : \mathbb{R}^{n_1} \to \mathbb{R}^m$, and $H : \mathbb{R}^{n_2} \to \mathbb{R}^m$ be such that $f(x, y) = g(x) + h(y)$ and $F(x, y) = G(x) + H(y)$ for all $(x, y) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$. Because we are specifically interested in the case of linear functions, we also introduce a right-hand side $b \in \mathbb{R}^m$, as is standard for the problems involving linear functions. We then obtain the new form of general optimization problem

$$\min_{x \in X,\ y \in Y} \{g(x) + h(y) \mid G(x) + H(y) \ge b\} \tag{GP-AS}$$

that we consider in the rest of the paper.

2.2.1 Projection and the Value Function

A reformulation of (GP-AS) analogous to (GP-Proj), obtained upon projecting into the space of the first-stage variables, is

$$\min_{x \in X} \{g(x) + \phi_{VF}(b - G(x))\}, \tag{GP-VF}$$

where

$$\phi_{VF}(\beta) = \min_{y \in \mathbb{R}^{n_2}_+} \{h(y) \mid y \in \mathcal{P}_2(\beta) \cap Y\} \quad \forall \beta \in \mathbb{R}^m, \tag{SP-VF}$$

and

$$\mathcal{P}_2(\beta) = \left\{ y \in \mathbb{R}^{n_2}_+ \;\middle|\; H(y) \geq \beta \right\} \quad \forall \beta \in \mathbb{R}^m$$

is a parametric family of polyhedra containing points satisfying the second-stage feasibility conditions. By convention, $\phi_{VF}(\beta) = +\infty$ for $\beta \in \mathbb{R}^m$ if $\mathcal{P}_2(\beta) \cap Y = \emptyset$, and $\phi_{VF} = -\infty$ if the second-stage problem is unbounded.

As opposed to the earlier-defined function ϕ_{SS}, which was parameterized on the first-stage solution, ϕ_{VF} is parameterized on the right-hand side of the associated second-stage optimization problem (which is in turn determined by the first-stage solution). The second-stage optimization problem, analogous to the earlier defined subproblem (GP-SP), is to evaluate ϕ_{VF} at a specific right-hand side. In the context of the framework described in Figure 1, ϕ_{VF} would be evaluated in iteration k of the algorithm at the right-hand side $\beta^k = b - G(x^k)$.

Those readers familiar with the more general duality theory associated with mixed integer linear optimization problems (see, e.g., Nemhauser and Wolsey [1988], Güzelsoy and Ralphs [2007]) will recognize ϕ_{VF} as the *value function* of the second-stage problem. The value function of the second-stage problem and the associated dual problem are a crucial element of the framework in the additively separable case, so we now briefly review these basic concepts.

2.2.2 General Duality and Dual Functions

In general, the value function of an optimization problem returns the optimal objective value for a given right-hand side vector. As in the general framework from Section 2.1, we form the master problem by replacing the value function with a function that bounds it from below. Such functions are known as *dual functions* and arise naturally as the solutions to a *general dual problem* that captures the essential role of duality in solution algorithms for optimization problems. For a particular right-hand side $\hat{\beta} \in \mathbb{R}^m$, the general dual problem

$$\max_{D \in \Upsilon^m} \left\{ D(\hat{\beta}) \;\middle|\; D(\beta) \leq \phi_{VF}(\beta) \quad \forall \beta \in \mathbb{R}^m \right\} \tag{SP-VF-GD}$$

associated with (GP-VF) is an optimization problem over a class $\Upsilon^m \subseteq \{\upsilon \mid \upsilon : \mathbb{R}^m \to \mathbb{R}\}$ of real-valued functions. The objective of the dual problem is to construct a function that bounds the value function from below and for which the bound is as strong as possible at $\hat{\beta}$. As such, we define a *(strong) dual function* as follows.

Definition 3 (**Dual Function**). *A dual function* $D : \mathbb{R}^m \to \mathbb{R} \cup \{\pm\infty\}$ *is one that satisfies* $D(\beta) \leq \phi_{VF}(\beta)$ *for all* $\beta \in \mathbb{R}^m$*. It is called* strong *at* $\hat{\beta} \in \mathbb{R}^m$ *if* $D(\hat{\beta}) = \phi_{VF}(\hat{\beta})$*.*

The dual problem itself is called *strong* if Υ^m is guaranteed to contain a strong dual function. As long as the value function itself is real-valued[1] and is a member of Υ^m, then the dual will be strong, since the value function itself is an optimal solution of (SP-VF-GD).

Exact solution algorithms that produce certificates of optimality typically do it by providing a primal solution, which certifies an upper bound, and a dual function (solution to (SP-VF-GD)), which certifies a lower bound. When these bounds are equal, the combination provides the required certificate of optimality. Dual functions can be obtained in various ways, but one obvious way to construct them is to consider the value function of a relaxation of the problem. Most solution algorithms for linear and mixed integer linear optimization problems work by iteratively constructing such a dual function.

The connection between the general dual and Benders' framework should be clear. The strong dual function constructed as a certificate of optimality when solving the subproblem (evaluating ϕ_{VF}) is a function that can be directly used in strengthening the global dual function that defines the current master problem. In fact, it is useful to think of the subproblem not as that of evaluating ϕ_{VF}, but rather of solving a dual problem of the form (SP-VF-GD) to obtain a strong dual function, which is what we actually need for forming the master problem in the next iteration.

2.2.3 Overall Algorithm

The overall method is largely similar to that described in Figure 1 when applied to problems with additively separable functions. We relax the reformulation (GP-VF) to obtain a master problem

$$\begin{aligned} \min_{x \in X} \; & g(x) + z \\ \text{s.t. } & z \geq \underline{\phi}_{VF}\left(b - G(x)\right), \end{aligned} \tag{MP-VF}$$

in which the value function ϕ_{VF} is replaced by a dual function. Solving this master problem in iteration k yields a solution (x^k, z^k) and a lower bound $g(x^k)+z^k$. We then evaluate ϕ_{VF} at $b-G(x^k)$ to obtain a dual function $\underline{\phi}^k_{VF}$ strong at $b-G(x^k)$ and an upper bound $g(x^k)+\phi_{VF}(b-G(x^k))$. This dual function is combined with previously produced such functions to obtain a global dual function that is strong at all previous iterates, ensuring eventual convergence under the same conditions as in Theorem 1.

Naturally, the exact form and structure of the dual functions involved is crucially important to the tractability of the the overall algorithm, as we do need a method of reformulating and solving the master problem in each iteration. In the cases discussed in this paper, the dual function takes on relatively simple forms. For linear optimization problems, the value function is convex and there is always a strong dual function that is a simple linear function. This linear function is an optimal solution to the usual LP dual, which is a subgradient of the LP value function.

[1] When the value function is not real-valued everywhere, we have to show that there exists a real-valued function that coincides with the value function whenever the value function is real-valued and is itself real-valued everywhere else, but is still a feasible dual function (see Wolsey [1981a]).

In the case of mixed integer linear optimization, dual functions can be obtained as a by-product of a branch-and-bound algorithm. Roughly speaking, the lower bound produced by branch-and-bound algorithm is the minimum of lower bounds produced for the individual subproblems associated with the leaf nodes of the branch-and-bound tree. Thus, the overall dual function is the minimum of dual functions for these subproblems. In the MILP case, the subproblem dual functions utilized are affine functions derived from the dual of the LP relaxation of the subproblem associated with a given node. Thus, in the simplest case, the dual function is the minimum of affine functions.

This method of constructing dual functions from branch-and-bound trees can be extended to virtually any problem that can be solved by the branch-and-bound framework. The lower bound arising from the branch-and-bound tree is the minimum of lower bounds on individual subproblems, which are typically (but not always) derived as dual functions of convex relaxations. The overall dual function is thus a minimum of dual functions for individual leaf nodes, as in the MILP case. In Sections 3.2.1 and 4.2 below, we describe in detail the application of this principle to the derivation of dual functions on the MILP and lexicographic MILP cases.

3 Applications From the Literature

In this section, we describe the application of the generalized framework presented in Section 2 to derive algorithms already existing in the literature. We describe these applications here to emphasize their commonality, and to provide concrete examples in settings in which the application of the principles is relatively straightforward and the abstractions reduce to familiar algorithmic concepts.

3.1 Linear Optimization Problems

We begin by considering the application of Benders' framework to the standard linear optimization problem (LP)

$$\min \left\{ cx + dy \;\middle|\; Ax + Gy \geq b,\ (x, y) \in \mathbb{R}^{n_1}_{+} \times \mathbb{R}^{n_2}_{+} \right\}, \tag{LP}$$

where $A \in \mathbb{Q}^{m \times n_1}$, $G \in \mathbb{Q}^{m \times n_2}$, $b \in \mathbb{Q}^m$, $c \in \mathbb{Q}^{n_1}$ and $d \in \mathbb{Q}^{n_2}$. This problem is the special case of (GP-AS) in which $g(x) = cx$, $h(y) = dy$, $G(x) = Ax$, and $H(y) = Gy$ for all $x \in X = \mathbb{R}^{n_1}_{+}$ and $y \in Y = \mathbb{R}^{n_2}_{+}$.

Projection. Projecting into the space of the first-stage variables, we obtain the reformulation

$$\min_{x \in \mathbb{R}^{n_1}_{+}} \{cx + \phi_{LP}(b - Ax)\}, \tag{LP-Proj-VF}$$

where

$$\phi_{LP}(\beta) = \min \left\{ dy \mid Gy \geq \beta,\ y \in \mathbb{R}^{n_2}_{+} \right\} \quad \forall \beta \in \mathbb{R}^m, \tag{LP-SP}$$

is the value function of the second-stage optimization problem, which is an LP. This reformulation is nothing more than the instantiation of the reformulation (GP-VF) in the context of (LP).

Value functions are well-studied and well-understood in the linear optimization case (see, e.g., Bertsimas and Tsitsiklis [1997] for details). The structure of ϕ_{LP} arises from that of the feasible region

$$\mathcal{D} = \{\eta \in \mathbb{R}^m_+ \mid G\eta \le d\}$$

of the standard LP dual of the second-stage problem, which is the LP

$$\max_{\eta \in \mathcal{D}} \eta(b - A\hat{x}) \tag{LPD}$$

when the first-stage solution is $\hat{x} \in \mathbb{R}^{n_1}_+$. Although it is not obvious, this dual problem is precisely equivalent to the general dual (SP-VF-GD) in the LP case. This can be seen by noting the constraints ensure that the dual solution is a subgradient of ϕ_{LP} and hence represents a (linear) dual function. By noting that the above maximum can be taken over the set $\mathcal{E}$ of extreme points of $\mathcal{D}$, assuming $\mathcal{D}$ is bounded, it is easy to derive that

$$\phi_{LP}(\beta) = \max_{\eta \in \mathcal{E}} \{\eta\beta\} \quad \forall \beta \in \mathbb{R}^m. \tag{LP-VF-Struct}$$

That is, the value function is the maximum of linear functions corresponding to members of $\mathcal{E}$. Although this function is convex and nicely structured, the cardinality of $\mathcal{E}$ is exponential in general, so enumerating them is impractical. The global dual function we use to construct the master problem is thus formed from a small collection of these extreme points, as described next.

Master Problem. In accordance with the principles described earlier, we form the master problem by replacing ϕ_{LP} in (LP-Proj-VF) with a global dual function $\underline{\phi}_{LP}$ that is the maximum of the strong dual functions produced in each iteration of the algorithm. In this context, the strong dual functions produced at each iteration are the linear functions associated with solutions to the dual (LPD) of the second-stage problem. This results in the master problem

$$\begin{aligned} \min\ & cx + z \\ \text{s.t. } & z \ge \underline{\phi}^i_{LP}(b - Ax) = \eta^i(b - Ax), \quad 1 \le i \le k \\ & x \in \mathbb{R}^{n_1}_+, \end{aligned} \tag{LP-MP}$$

in iteration k, where $\eta^i \in \mathcal{E}$ is an optimal solution of (LPD) in iteration i. Note that if the optimal solution to the master problem in iteration k is (x^k, z^k), then we have

$$z^k = \underline{\phi}_{LP}(b - Ax^k) = \max_{1 \le i \le k} \eta^i(b - Ax^k),$$

as desired, and this master problem is the equivalent to (MP-VF) in this context.

Overall Algorithm. In iteration k of the algorithm, we begin by solving a master problem (LP-MP) to obtain its optimal solution x^k. The subproblem is then to evaluate ϕ_{LP} at $\beta^k = b - Ax^k$ by solving the dual (LPD) to obtain $\eta^k \in \mathcal{E}$. By defining $\underline{\phi}^k_{LP}(\beta) = \eta^k\beta$ for all $\beta \in \mathbb{R}^m$, we obtain that $\underline{\phi}^k_{LP}$ is a dual function for ϕ_{LP} that is strong at β^k.

The overall method is then to add one constraint of the form

$$z \ge \eta^k(b - Ax) \tag{Opt-Cut}$$

to the master problem in each iteration k in which (LPD) has a finite optimum. In case (LPD) in iteration k is unbounded, then x^k is not a member of the projection of the feasible region (defined as in (GP-FR-Proj)) and we instead add a constraint of the form

$$0 \geq \sigma^k(b - Ax), \tag{Feas-Cut}$$

where σ^k is the extreme ray of $\mathcal{D}$ that proves infeasibility of the second-stage problem.

Observe that the master problem, the subproblem, and the optimality constraints described in this section are identical to the corresponding components in a classical Benders' decomposition algorithm for LPs. Conventionally, in the context of LPs, the optimality constraints are referred to as Benders' cuts with (Opt-Cut) being *Benders' optimality cut* and (Feas-Cut) being *Benders' feasibility cut*. Next, we discuss how to further generalize to the class of 2SSMILPs.

3.2 Two-Stage Stochastic Mixed Integer Linear Optimization Problems

The case of 2SSMILPs generalizes the case discussed in the previous section in two important ways. First, we introduce stochasticity, which is modeled by specifying a finite number of possible scenarios in the second stage, resulting in a block-structured constraint matrix overall. This generalization on its own is relatively straightforward and results in the method known in the literature as the L-shaped method for solving stochastic linear optimization problems with recourse [Van Slyke and Wets, 1969]. However, we also wish to allow integer variables into the second stage. Although this doesn't require any modification of the framework itself, it results in a more complex structure for the value function of the second-stage problem and hence, a more complex reformulation for the master problem. We now summarize an algorithm for solving 2SSMILPs that was originally developed by Hassanzadeh and Ralphs [2014a].

To model the stochasticity, we introduce a random variable U over an outcome space Ω representing the set of possible scenarios for the second-stage problem. We assume that U is discrete, i.e., that the outcome space Ω is finite so that $\omega \in \Omega$ represents which of the finitely many scenarios is realized. In practice, this assumption is not very restrictive, as one can exploit any algorithm for the case in which Ω is assumed finite to solve cases where Ω is not (necessarily) finite by utilizing a technique for discretization, such as *sample average approximation* (SAA) [Shapiro, 2003].

Under these assumptions, a 2SSMILP is then a problem of the form

$$\begin{aligned} \min\ & cx + \mathbb{E}_{\omega \in \Omega}\left[d^2 y_\omega\right] \\ \text{s.t. }\ & A^1 x \geq b^1 \\ & G^2 y_\omega \geq b^2_\omega - A^2_\omega x \quad \forall \omega \in \Omega \\ & x \in X, y_\omega \in Y. \end{aligned} \tag{2SSMILP}$$

where $c \in \mathbb{Q}^{n_1}$, $d^2 \in \mathbb{Q}^{n_2}$, $A^1 \in \mathbb{Q}^{m_1 \times n_1}$, $G^2 \in \mathbb{Q}^{m_2 \times n_2}$, and $b^1 \in \mathbb{Q}^{m_1}$. $A^2_\omega \in \mathbb{Q}^{m_2 \times n_1}$ and $b^2_\omega \in \mathbb{Q}^{m_2}$ represent the realized values of the random input parameters in scenario $\omega \in \Omega$, i.e., $U(\omega) = (A^2_\omega, b^2_\omega)$. The first term in the objective function represents the immediate cost of implementation of the first-stage solution, while the second term is an expected cost over the set of possible future scenarios.

3.2.1 Projection

We now reformulate the problem by exploiting two important properties. First, since U is discrete, we may associate with it a discrete probability distribution defined by $p \in \mathbb{R}^{|\Omega|}$ such that $0 \le p_\omega \le 1$ and $\sum_{\omega\in\Omega} p_\omega = 1$, where p_ω represents the probability of the scenario $\omega \in \Omega$. This allows us to replace the expectation above with a finite sum. Second, we note that the second-stage problem has a natural block structure, so that the problem decomposes perfectly into $|\Omega|$ subproblems, which differ only in the right-hand side vector once the first-stage solution is fixed (this is one of the advantages of using this approach in this setting). Thus, when we project this problem as before into the space of the first-stage variables, we obtain the reformulation

$$\min \left\{ cx + \sum_{\omega\in\Omega} p_\omega \phi(b^2_\omega - A^2_\omega x) \;\middle|\; A^1 x \ge b^1,\ x \in X \right\}, \tag{2SSMILP-SE}$$

where the value function ϕ of the second-stage problem is defined by

$$\phi(\beta) = \min_{y\in\mathbb{R}^{n_2}_+} \left\{ d^2 y \;\middle|\; y \in \mathcal{P}_2(\beta) \cap Y \right\} \quad \forall \beta \in \mathbb{R}^{m_2}, \tag{MILP-VF}$$

and

$$\mathcal{P}_2(\beta) = \left\{ y \in \mathbb{R}^{n_2}_+ \;\middle|\; G^2 y \ge \beta \right\} \quad \forall \beta \in \mathbb{R}^{m_2}. \tag{MILP-VF-FR}$$

By convention, $\phi(\beta) = \infty$ if the feasible region $\{y \in \mathbb{R}^{n_2}_+ \mid y \in \mathcal{P}_2(\beta) \cap Y\}$ is empty, and $\phi(\beta) = -\infty$ if the second-stage problem associated with β is unbounded which results in $\phi = -\infty$ for all $\beta \in \mathbb{R}^{m_2}$. Note that the single second-stage value function that appeared in the analogous reformulation in Section 2.2 is replaced here with expected value of the value function across all scenarios, expressed as a weighted sum. Although each scenario results in a separate outcome, the evaluation of those scenario outcomes is done in principle using the single second-stage value function ϕ, which links the scenarios.

It is clear from (2SSMILP-SE) that solving (2SSMILP) (directly or iteratively) requires exploiting the structure of the MILP value function (MILP-VF), just as we exploited the structure of the LP value function in solving (LP-Proj-VF) in Section 3.1. Therefore, we now take a quick detour to discuss this structure and construction of a strong dual in the MILP case.

The MILP Value Function. The structure of the value function of an MILP is by now well-studied. Early foundational works include Johnson [1973, 1974], Blair and Jeroslow [1977, 1979a,b, 1982, 1984], Jeroslow [1978, 1979], Bachem and Schrader [1980], Bank et al. [1983], Blair [1995]. A number of later works extended these foundational results [Güzelsoy and Ralphs, 2007, Güzelsoy, 2009, Hassanzadeh and Ralphs, 2014b, Hassanzadeh, 2015]. What follows is a summary of results from these later works.

Let us first look at an example to get some intuition about the structure of ϕ.

Example 1 Consider the following parametric MILP, described by its associated value function, plotted in Figure 2.

$$\begin{aligned} \phi(\beta) = \quad & \min\ 2y_1 + 4y_2 + 3y_3 + 4y_4 \\ & \text{s.t. } 2y_1 + 5y_2 + 2y_3 + 2y_4 \ge \beta, \\ & y_1, y_2, y_3 \in \mathbb{Z}_+,\ y_4 \in \mathbb{R}_+ \end{aligned} \tag{MILP-Toy}$$

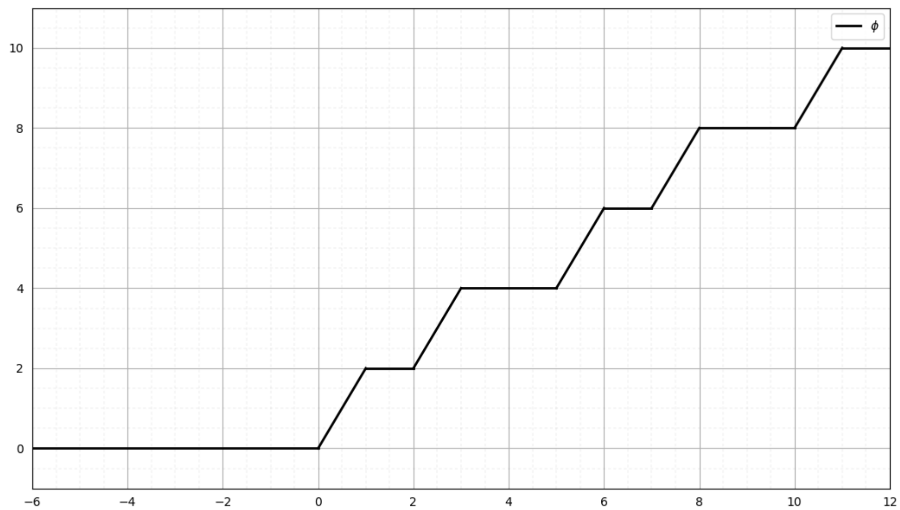


Figure 2: MILP value function ϕ (MILP-Toy)

□

The function shown in Figure 2 is observed to be piecewise linear, non-decreasing, non-convex, and non-concave. These are all properties of general value functions of the form (MILP-VF), but there are also several other important properties that are not evident from this simple example. In particular, the value function may be discontinuous, but is always lower semi-continuous and also subadditive.

Another important property of ϕ that is evident from Figure 2 is that its epigraph is the union of a set of convex radial cones. These cones are translations of the epigraph of the value function of a single parametric LP, the so-called *continuous restriction* of the given MILP (defined later in Section 4.2) resulting from fixing the integer variables. Hassanzadeh and Ralphs [2014b] further proved that the value function can be described within any bounded region by specifying a finite set of points of *strict local convexity* of ϕ, which are the locations of the extreme points of these radial cones (assuming the epigraph of the LP value functions is a pointed cone). This resulted in a finite discrete representation of ϕ (see Hassanzadeh and Ralphs [2014b] for additional details and formal results).

Although there exist effective algorithms for evaluating ϕ for a single fixed right-hand side $\hat{\beta}$ (e.g., any method for solving the associated MILP), it is difficult to explicitly construct the entire function because this evidently requires solution of a sequence of MILPs. Algorithms for evaluating ϕ at a right-hand side $\hat{\beta}$, such as branch-and-bound algorithm, do, however, produce information about its structure beyond the single value $\phi(\hat{\beta})$. This information comes in the form of a dual function that is strong at $\hat{\beta}$. We next describe the form and structure of these dual functions.

Dual Functions. Dual functions can be obtained from virtually any algorithm that produces a primal-dual proof of optimality. We focus here on dual functions from the branch-and-bound algorithm, which is the most widely used solution method for solving MILPs. We refer the reader to Güzelsoy [2009], Güzelsoy and Ralphs [2007] for an overview of other methods. Wolsey [1981b] was the first to propose that dual functions could be extracted from branch-and-bound trees, as described in the following result.

Theorem 2. *[Wolsey, 1981b] Let $\hat{\beta} \in \mathbb{R}^{m_2}$ be such that $\phi(\hat{\beta}) < \infty$ and suppose T is the set of indices of leaf nodes of a branch-and-bound tree resulting from evaluation of $\phi(\hat{\beta})$. Then there exists a dual function $\underline{\phi} : \mathbb{R}^{m_2} \to \mathbb{R} \cup \{\pm\infty\}$ of the form*

$$\underline{\phi}(\beta) = \min_{t \in T}(\beta\eta^t + \alpha^t) \quad \forall \beta \in \mathbb{R}^{m_2}, \tag{MILP-DF}$$

where $\eta^t \in \mathbb{R}^{m_2}$ is an optimal solution to the dual of the LP relaxation associated with node t and $\alpha^t \in \mathbb{R}$ is the product of the optimal reduced costs and variable bounds of this LP relaxation. Further, $\underline{\phi}$ is strong at $\hat{\beta}$, i.e., $\underline{\phi}(\hat{\beta}) = \phi(\hat{\beta})$.

The interpretation of the function $\underline{\phi}$ in (MILP-DF) is conceptually straightforward. The solution to the LP relaxation of node t of the branch-and-bound tree yields the standard LP dual function $\left(\beta\eta^t + \alpha^t\right)$, which bounds the optimal value of the relaxation problem associated with that node. The overall lower bound yielded by the tree is then 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. Finally, we obtain $\underline{\phi}$ by interpreting the optimal solution to the dual of the LP relaxation in each node as a function.

One subtle point we should address further has to do with the infeasible nodes. In the above discussion, we simply appealed to the existence of a dual solution that could be used to construct an appropriate dual function at each node. In practice, we need to be able to obtain such dual solutions in a practical way. Consider again a node $t \in T$ whose LP relaxation is infeasible. One method of obtaining an appropriate dual solution is to let $(\sigma^t, \underline{\sigma}^t, \overline{\sigma}^t)$ be an extreme ray of the dual feasible region that proves the infeasibility of the primal LP relaxation and let $(\tilde{\eta}^t, \tilde{\underline{\eta}}^t, \tilde{\overline{\eta}}^t)$ be a dual feasible solution generated just prior to discovery of the dual ray. In these tuples, the last two elements correspond to reduced costs of the LP relaxation, which can also be interpreted as the dual multipliers on the bound constraints on the variables. By adding an appropriately chosen scalar multiple of the ray to this dual solution, we obtain a second dual solution with the desired property that we can use in Theorem 2. More formally, let $\lambda \in \mathbb{R}^+$ be a given scalar and consider

$$(\hat{\eta}^t, \hat{\underline{\eta}}^t, \hat{\overline{\eta}}^t) = (\tilde{\eta}^t, \tilde{\underline{\eta}}^t, \tilde{\overline{\eta}}^t) + \lambda(\sigma^t, \underline{\sigma}^t, \overline{\sigma}^t).$$

By choosing λ large enough, we obtain a feasible dual solution with a bound smaller than the optimum. 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.

In principle, stronger dual functions can be obtained. For example, stronger functions can be constructed from the branch-and-bound tree by considering non-leaf nodes, suboptimal dual solutions

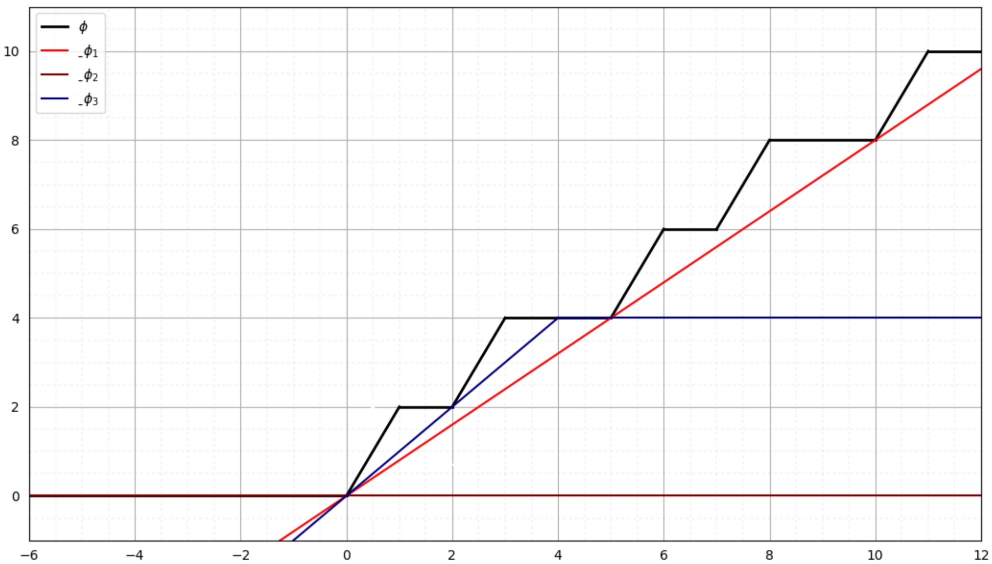


Figure 3: Dual functions for (MILP-Toy)

arising during the solution process, full LP value function at each leaf node t instead of a single hyperplane $\beta\eta^t + \alpha^t$, etc. Further details on these methods are mentioned in Güzelsoy and Ralphs [2007], Hassanzadeh and Ralphs [2014a].

Example 2 Figure 3 shows the dual functions obtained upon applying the result in Theorem 2 to the MILP (MILP-Toy). We solve this MILP with three values of the right-hand side β.

- $\beta = 5$: There is only one node in the associated branch-and-bound tree with the optimal dual solution $\eta = 0.8$, $\underline{\eta} = (0.4, 0, 1.4, 2.4)$, and $\overline{\eta} = (0, 0, 0, 0)$. This results in the dual function

$$\underline{\phi}_1(\beta) = 0.8\beta \quad \forall \beta \in \mathbb{R}.$$

- $\beta = 0$: There is still only one node in the tree with the optimal dual solution $\eta = 0$, $\underline{\eta} = (2, 4, 3, 4)$, and $\overline{\eta} = (0, 0, 0, 0)$. This results in the dual function

$$\underline{\phi}_2(\beta) = 0 \quad \forall \beta \in \mathbb{R}.$$

- $\beta = 2$: There are three nodes in the tree, i.e., one root node and two leaf nodes resulting from the branching disjunction $y_2 \leq 0 \vee y_2 \geq 1$. The optimal dual solution and the resulting dual function $\beta\eta^t + \alpha^t$ at each leaf node t are mentioned in Table 1. This results in the dual function

$$\underline{\phi}_3(\beta) = \min\{\beta, 4\} \quad \forall \beta \in \mathbb{R}.$$

t	Branching constraint	η^t	$\underline{\eta}^t$	$\overline{\eta}^t$	$\beta\eta^t + \alpha^t$
1	$y_2 \leq 0$	1	(0, 0, 1, 2)	(0, -1, 0, 0)	β
2	$y_2 \geq 1$	0	(2, 4, 3, 4)	(0, 0, 0, 0)	4

Table 1: Data for construction of the dual function from the branch-and-bound tree in Example 2

Naturally, as in the formulation of the master problem, the value function approximation can be improved by taking the maximum of multiple dual functions strong at different right-hand sides. In the above example, the dual function $\max\{\underline{\phi}_1(\beta), \underline{\phi}_2(\beta), \underline{\phi}_3(\beta)\}$ is already seen to be a reasonable approximation of the full value function. □

3.2.2 Master Problem

By exploiting the specific structure of the dual functions described in the previous section, we can straightforwardly adapt the algorithmic framework from Section 2 to obtain an algorithm for solving (2SSMILP-SE) similar to that derived by the authors in Hassanzadeh and Ralphs [2014a].

As we mentioned before, the second-stage problem has a natural block structure and decomposes into $|\Omega|$ independent subproblems, which differ only in the right-hand side. In iteration k of the algorithm, evaluating the second-stage value function overall involves evaluating ϕ at $|\Omega|$ different points, one for each scenario. Thus, the subproblem in iteration k is to evaluate

$$\begin{aligned}\phi(b^2_\omega - A^2_\omega x^k) = \min\ & d^2 y \\ \text{s.t. } & G^2 y \geq b^2_\omega - A^2_\omega x^k \\ & y \in Y\end{aligned} \tag{2SSMILP-SP}$$

for all $\omega \in \Omega$, where x^k is an optimal solution to the master problem in the current iteration. The result is a scenario dual function $\underline{\phi}^k_\omega$ of the form (MILP-DF) for each $\omega \in \Omega$. Introducing auxiliary variables z_ω for each scenario, as in previous reformulations, we obtain the master problem in iteration k as

$$\begin{aligned}\min\ & cx + \sum_{\omega\in\Omega} p_\omega z_\omega \\ \text{s.t. } & A^1 x \geq b^1 \\ & z_\omega \geq \max_{1\leq i\leq k} \underline{\phi}^i_\omega(b^2_\omega - A^2_\omega x) \quad \forall \omega \in \Omega \\ & x \in X.\end{aligned} \tag{2SSMILP-MP}$$

Because each scenario dual function is the minimum of a collection of affine functions, the overall master problem can be eventually reformulated as an MILP by introducing additional binary variables (see Hassanzadeh and Ralphs [2014a] for details).

3.2.3 Overall Algorithm

Putting this all together, in each iteration k, a master problem of the form (2SSMILP-MP) is solved to obtain its optimal solution $(x^k, \{z^k_\omega\}_{\omega\in\Omega})$ and a lower bound. Following that, one scenario subproblem is solved for each $\omega \in \Omega$, which consists of evaluating $\phi(b^2_\omega - A^2_\omega x^k)$ using a branch-and-bound algorithm. The result is a strong dual function (MILP-DF) for each scenario, as well as an overall upper bound. If the upper and lower bounds are equal, then we're done. Otherwise, the dual functions are fed back into the master problem and the method is iterated until the upper and lower bounds converge.

4 Mixed Integer Bilevel Linear Optimization Problems

We now move on to a detailed discussion of the application of the generalized Benders' principle to MIBLPs, the class of problems in which we are most interested. As described earlier, MIBLPs are two-stage MMILPs in which the variables at each stage are conceptually controlled by different DMs with different objective functions.

MIBLPs model problems in game theory, specifically the *Stackelberg games* introduced by Von Stackelberg [1934]. Bilevel optimization problems in the form presented here were formally introduced and the term was coined in the 1970s by Bracken and McGill [Bracken and McGill, 1973], but computational aspects of such optimization problems have been studied since at least the 1960s (see, e.g., Wollmer [1964]). Most of the initial research was limited to continuous bilevel linear optimization problems containing only continuous variables and linear constraints in both the stages.

Study of bilevel optimization problems containing integer variables and algorithms for solving them is generally acknowledged to have been initiated by Moore and Bard [1990], who discussed the computational challenges of solving such problems and suggested one of the earliest algorithms, a branch-and-bound algorithm that converges to an optimal solution if all first-stage variables are integer variables or all second-stage variables are continuous variables.

Since then, many works have focused on special cases, such as having only binary integer variables in the first stage or only continuous variables in the second stage. It is only in the past decade or so that study of exact algorithms for the general case in which there are both continuous and general integer variables in both stages has been undertaken. Table 2 provides a timeline of the main developments in the evolution of such exact algorithms, while indicating the types of variables supported in both the first and second stages (C indicates continuous, B indicates binary, and G indicates general integer).

4.1 Formulation

To state the class of problems formally, we introduce a type of constraint that doesn't seem to naturally fit the paradigm of a standard mathematical optimization problem. In addition to the usual linear constraints, we have a constraint that requires the second-stage solution to be optimal with respect to a problem that is parametric in the first-stage solution. The formulation including

Citation	Stage 1 Variable Types	Stage 2 Variable Types
Wen and Yang [1990]	B	C
Bard and Moore [1992]	B	B
Faísca et al. [2007]	B, C	B, C
Saharidis and Ierapetritou [2008]	B, C	B, C
Garcés et al. [2009]	B	C
DeNegre and Ralphs [2009], DeNegre [2011]	G	G
Köppe et al. [2010]	G or C	G
Baringo and Conejo [2012]	B, C	C
Xu and Wang [2014]	G	G, C
Zeng and An [2014]	G, C	G, C
Caramia and Mari [2015]	G	G
Caprara et al. [2016]	B	B
Hemmati and Smith [2016]	B, C	B, C
Tahernejad et al. [2016]	G, C	G, C
Wang and Xu [2017]	G	G
Lozano and Smith [2017]	G	G, C
Fischetti et al. [2018], Fischetti et al. [2017]	G, C	G, C

Table 2: Evolution of algorithms for bilevel optimization

this constraint, as it usually appears in the literature on bilevel optimization, is

$$
\begin{aligned}
\min\ & cx + d^1 y \\
\text{s.t. } & A^1 x + G^1 y \geq b^1 \\
& x \in X \\
& y \in \arg\min \{ d^2 \check{y} \\
& \quad \text{s.t. } G^2 \check{y} \geq b^2 - A^2 x \\
& \quad \check{y} \in Y \},
\end{aligned}
\tag{MIBLP}
$$

where $A^1 \in \mathbb{Q}^{m_1 \times n_1}$, $G^1 \in \mathbb{Q}^{m_1 \times n_2}$, $b^1 \in \mathbb{Q}^{m_1}$, $A^2 \in \mathbb{Q}^{m_2 \times n_1}$, $G^2 \in \mathbb{Q}^{m_2 \times n_2}$, $b^2 \in \mathbb{Q}^{m_2}$, $c \in \mathbb{Q}^{n_1}$, $d^1 \in \mathbb{Q}^{n_2}$, and $d^2 \in \mathbb{Q}^{n_2}$. Note that the above-mentioned parametric problem is nothing but the evaluation of $\phi(b^2 - A^2 x)$, where ϕ is the MILP value function (MILP-VF).

Underlying the above formulation are a number of assumptions. First, there is an implicit assumption that whenever the evaluation of $\phi(b^2 - A^2 x)$ yields multiple optimal solutions, the one that is most advantageous for the first-stage DM is chosen. This form of MIBLP is known as the *optimistic* case and is just one of several variants. The *pessimistic* variant, for example, is one in which the second-stage solution chosen is always the one *least* advantageous for the first-stage DM. It should also be pointed out that we explicitly allow the second-stage variables in the constraints $A^1 x + G^1 y \geq b^1$. This is rather non-intuitive but there are applications for which this is a necessary element. We now define so-called *linking variables*.

Definition 4 (Linking Variables). *Linking variables are the first-stage variables whose indices* *are in the set*

$$L = \left\{ i \in \{1, \dots, n_1\} \mid A^2_i \neq 0 \right\},$$

where A^2_i denotes the i^{th} column of A^2.

Assumption 1. *All linking variables are integer variables.*

This assumption is required to ensure the existence of an optimal solution for the given MIBLP. The optimal solution may not be attainable when there are linking variables that are continuous and second-stage variables that are integer [Moore and Bard, 1990, Vicente et al., 1996, Köppe et al., 2010].

Assumption 2. *All first-stage variables are linking variables.*

Since we focus on optimistic bilevel problems, all non-linking variables can simply be moved to the second stage without loss of generality. This assumption is made primarily for ease of exposition, nevertheless results in a mathematically equivalent MIBLP, despite the inconsistency with the intent of the original model. In combination with Assumption 1, this assumption implies that all first-stage variables are integer variables, i.e., $n_1 = r_1$.

Assumption 3. *The set*

$$\{(x, y) \in \mathbb{R}^{n_1}_+ \times \mathbb{R}^{n_2}_+ \mid y \in \mathcal{P}_1(b^1 - A^1x) \cap \mathcal{P}_2(b^2 - A^2x)\}$$

is bounded, where

$$\mathcal{P}_1(\beta^1) = \left\{ y \in \mathbb{R}^{n_2}_+ \;\middle|\; G^1 y \geq \beta^1 \right\}$$

and $\mathcal{P}_2(\beta^2)$ (as defined in (MILP-VF-FR)) represent families of polyhedra consisting of all points satisfying $G^1 y \geq \beta^1$ and $G^2 y \geq \beta^2$ for given right-hand sides $\beta^1 \in \mathbb{R}^{m_1}$ and $\beta^2 \in \mathbb{R}^{m_2}$. Assumption 3 is made to avoid uninteresting cases involving unboundedness, but is easy to relax in practice.

Assumption 4. *For all $x \in \mathbb{R}^{n_1}$, we have*

$$\phi(b^2 - A^2x) > -\infty,$$

or, equivalently

$$\{r \in \mathbb{R}^{n_2}_+ \mid G^2 r \geq 0, d^2 r < 0\} = \emptyset.$$

Assumption 4 is also made to avoid uninteresting cases involving unboundedness. Observe that in the case $\phi(b^2 - A^2x) = -\infty$ for a given value of x, then $\phi(b^2 - A^2x) = -\infty$ for all values of x. Note that Assumptions 2-4 can be relaxed in practice.

4.2 Projection

We now apply the familiar operations of Benders' framework to the formulation (MIBLP). Upon projecting into the space of the first-stage variables, we obtain the reformulation

$$\min_{x \in X} \left\{ cx + \rho(b^1 - A^1x,\ b^2 - A^2x) \right\}, \tag{MIBLP-ReF}$$

where ρ is the second-stage *reaction function*, defined as

$$\rho(\beta^1, \beta^2) = \min\left\{d^1 y \;\middle|\; y \in \mathcal{P}_1(\beta^1),\ y \in \arg\min\{d^2 \check{y} \mid \check{y} \in \mathcal{P}_2(\beta^2) \cap Y\}\right\}. \tag{ReF}$$

Although the reaction function appears at first to be the value function of a general bilevel optimization problem, it is actually the value function of a lexicographic MILP and encodes the part of the first-stage DM's objective function that depends on the response of the second-stage DM to a given first-stage solution. In the next part of this section, we examine its properties and structure before discussing how to construct associated dual functions.

Reaction Function. As with all value functions, $\rho(\beta^1, \beta^2) = \infty$ for a given $(\beta^1, \beta^2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ if either $\{y \in \mathbb{R}^{n_2}_+ \mid y \in \mathcal{P}_1(\beta^1) \cap \mathcal{P}_2(\beta^2) \cap Y\} = \emptyset$ or $\phi(\beta^2) = -\infty$ (which cannot happen under Assumption 4), and $\rho(\beta^1, \beta^2) = -\infty$ for all $(\beta^1, \beta^2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ if the lexicographic MILP is itself unbounded, i.e., $\{r \in \mathbb{R}^{n_2}_+ \mid G^1 r \geq 0, G^2 r \geq 0, d^1 r < 0\} \neq \emptyset$.

Observe that our initial definition (ReF) of the reaction function has embedded within it exactly the kind of optimality constraint we tried to eliminate by using projection to reformulate the given MIBLP. Using the MILP value function ϕ (MILP-VF), we can use a similar technique to again re-define the reaction function equivalently as

$$\rho(\beta^1, \beta^2) = \min\left\{d^1 y \;\middle|\; y \in \mathcal{P}_1(\beta^1) \cap \mathcal{P}_2(\beta^2) \cap Y, d^2 y \leq \phi(\beta^2)\right\}. \tag{ReF-VF}$$

Although ϕ is a nonlinear function in general, $\phi(\beta^2)$ is a constant for a fixed value of β^2. Hence, for a given value of (β^1, β^2), evaluating $\rho(\beta^1, \beta^2)$ is equivalent to solving an MILP.

We now illustrate the structure of the reaction function with a simple example. Although its structure is combinatorially more complex than that of the MILP value function, it nevertheless also has a piecewise polyhedral structure. We do not provide formal results concerning the structure and properties of the reaction function here, but these can be derived by application of techniques similar to those used to derive the structure of the MILP value function.

Example 3 Consider the following reaction function arising from an MIBLP with $G^1 = 0$.

$$\begin{aligned} \rho(\beta) = \min \quad & -y_1 + y_2 - 5y_3 + y_4 \\ (y_1, y_2, y_3, y_4) \in \arg\min \{ & 2\check{y}_1 + 4\check{y}_2 + 3\check{y}_3 + 4\check{y}_4 \\ & 2\check{y}_1 + 5\check{y}_2 + 2\check{y}_3 + 2\check{y}_4 \geq \beta \\ & \check{y}_1, \check{y}_2, \check{y}_3 \in \mathbb{Z}_+,\ \check{y}_4 \in \mathbb{R}_+\} \end{aligned} \tag{ReF-Toy}$$

This function can be reformulated as

$$\begin{aligned} \rho(\beta) = \min \quad & -y_1 + y_2 - 5y_3 + y_4 \\ & 2y_1 + 5y_2 + 2y_3 + 2y_4 \geq \beta \\ & 2y_1 + 4y_2 + 3y_3 + 4y_4 \leq \phi(\beta) \\ & y_1,\ y_2, y_3 \in \mathbb{Z}_+,\ y_4 \in \mathbb{R}_+. \end{aligned} \tag{ReF-Toy-VF}$$

using the MILP value function ϕ which is the same as for (MILP-Toy). The function ρ is plotted in Figure 4. It can be seen from this figure that the structure is quite complex, though still piecewise polyhedral. □

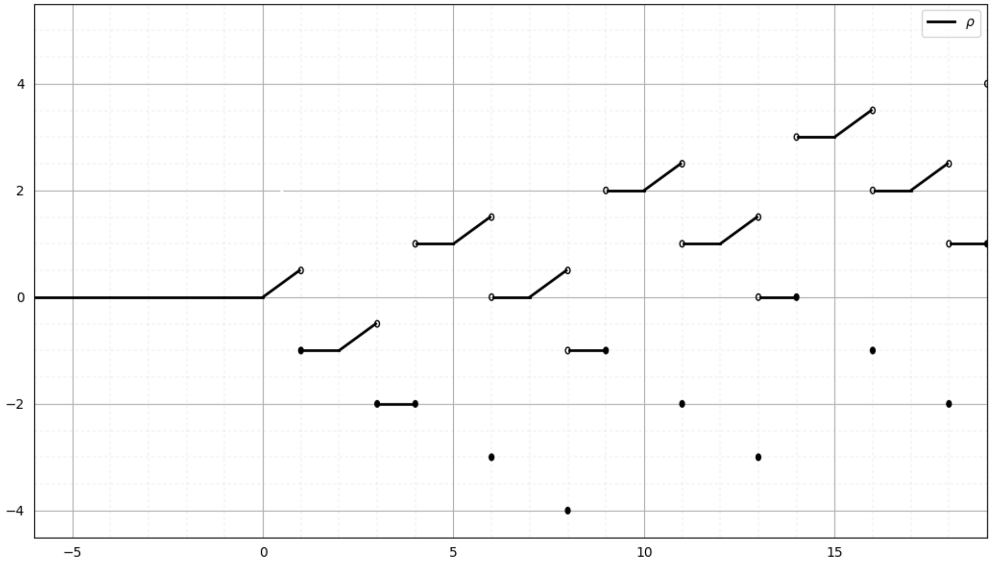


Figure 4: Reaction function ρ (ReF-Toy)

As usual, we do not have an exact description of ρ in general, so we cannot solve (MIBLP-ReF) directly and we replace ρ in (MIBLP-ReF) with a dual function $\underline{\rho}$. Following the earlier procedure, this dual function is taken to be the maximum of the strong dual functions $\underline{\rho}^k$ obtained by solving a subproblem in each iteration k. The resulting master problem in iteration k is

$$\begin{aligned} &\min\ cx + z \\ &\text{s.t. } z \geq \underline{\rho}^i(b^1 - A^1x,\ b^2 - A^2x), \quad 1 \leq i \leq k \\ &\quad x \in X. \end{aligned} \tag{MIBLP-MP}$$

As before, the subproblem in iteration k is to evaluate $\rho(b^1 - A^1x^k,\ b^2 - A^2x^k)$ for the solution x^k to (MIBLP-MP), in order to construct a dual function $\underline{\rho}^k$ strong at $(b^1 - A^1x^k,\ b^2 - A^2x^k)$. We next detail how this strong dual function is constructed.

Dual Functions. As we have already noted, the subproblem in iteration k is to evaluate the reaction function (ReF) for $(\beta^1, \beta^2) = (b^1 - A^1x^k, b^2 - A^2x^k)$, This problem is an MILP and we have the following theorem based on Theorem 2.

Theorem 3. *Let $(\hat{\beta}^1, \hat{\beta}^2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}$ be such that $\rho(\hat{\beta}^1, \hat{\beta}^2) < \infty$ and suppose T is the set of indices of leaf nodes of a branch-and-bound tree resulting from evaluation of $\rho(\hat{\beta}^1, \hat{\beta}^2)$. Then there exists a dual function $\underset{\sim}{\rho} : \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \to \mathbb{R} \cup \{\pm\infty\}$ of the form*

$$\underset{\sim}{\rho}(\beta^1, \beta^2) = \min_{t \in T} \left(\beta^1 \eta^{1,t} + \beta^2 \eta^{2,t} + \phi(\beta^2)\eta^{\phi,t} + \alpha^t \right) \quad \forall (\beta^1, \beta^2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2}, \tag{ReF-LBF}$$

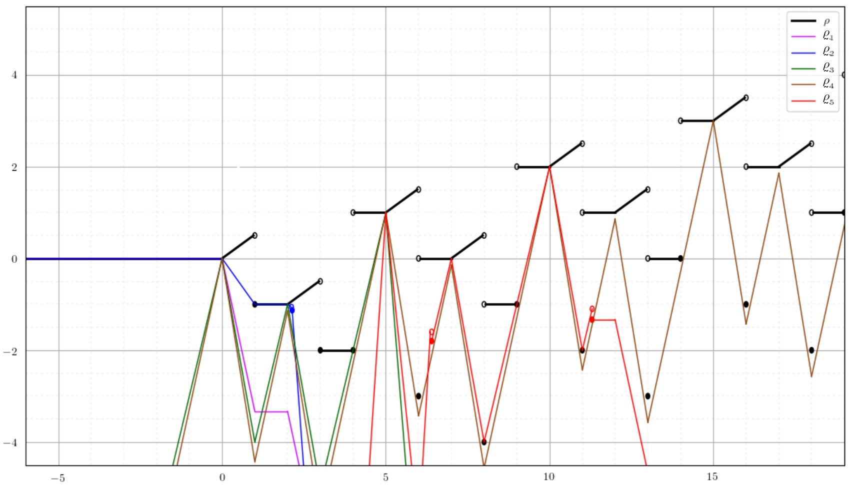


Figure 5: Dual functions for (ReF-Toy)

t	Branching constraint	$(\eta^{2,t}, \eta^{\phi,t})$	$\underline{\eta}^t$	$\overline{\eta}^t$	$\eta^{2,t}\beta + \eta^{\phi,t}\phi(\beta) + \alpha^t$
1	$y_2 \geq 2$	$(0, -\frac{5}{3})$	$(\frac{7}{3}, \frac{23}{3}, 0, \ \frac{7}{3})$	$(0, 0, 0, 0)$	$-\frac{5}{3}\phi(\beta) + \frac{46}{3}$
2	$y_2 \leq 1, \ y_1 \leq 0$	$(11, -9)$	(0, 0, 0, 15)	(-5, -18, 0, 0)	$11\beta - 9\phi(\beta) - 18$
3	$y_2 \leq 1, \ y_1 \geq 1, \ y_3 \leq 0$	$(3, -3.5)$	(0, 0, 0, 9)	(0, 0, -0.5, 0)	$3\beta - 3.5\phi(\beta)$
4	$y_2 \leq 1, \ y_1 \geq 1, \ y_3 \geq 1$	$(13, -16)$	(5, 0, 17, 39)	(0, 0, 0, 0)	$13\beta - 16\phi(\beta) + 22$

Table 3: Data for construction of the dual function in Example 4

where $(\eta^{1,t}, \eta^{2,t}, \eta^{\phi,t}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}$ is a dual feasible solution of the LP relaxation associated with node t, and $\alpha^t \in \mathbb{R}$ is the product of reduced costs and variable bounds of this LP relaxation. Further, this dual function is strong at $(\hat{\beta}^1, \hat{\beta}^2)$ if $\underset{\sim}{\rho}(\hat{\beta}^1, \hat{\beta}^2) = \rho(\hat{\beta}^1, \hat{\beta}^2)$.

The interpretation of this result is similar to the interpretation of Theorem 2. Let us look at an example now, depicting these functions.

Example 4 Figure 5 shows five dual functions obtained upon applying the result in Theorem 3 to the reaction function (ReF-Toy) for right-hand sides $\beta = 0, 1, 2, 5, 8$. As expected, these dual functions are piecewise polyhedral. For example, solving (ReF-Toy) with $\beta = 8$ as an equivalent MILP (after obtaining $\phi(8)$ at first) yields the dual information (dual solution and reduced costs) from leaf nodes of the branch-and-bound tree shown in Table 3.

This results in the dual function

$$\underset{\sim}{\rho}_5(\beta) = \min\left\{-\frac{5}{3}\phi(\beta) + \frac{46}{3}, 11\ \beta - 9\phi(\beta) - 18, 3\beta - 3.5\phi(\beta), 13\beta - 16\ \phi(\beta) + 22\right\},$$

containing the MILP value function ϕ. The remaining dual functions are obtained the same way. □

It is clear from Theorem 3 that construction of ϕ is required for construction of $\underset{\sim}{\rho}$ in (ReF-LBF). However, construction of ϕ is itself a difficult task and generally impractical. Further, the complex structure of ϕ makes the structure of $\underset{\sim}{\rho}$ highly complex. To work around these difficulties, we replace ϕ in (ReF-VF) with a *primal function* (defined formally below), which bounds the value function from above and is strong at the given right-hand side. This results in an alternative dual function, which we denote by $\underline{\rho}$ that is still strong at the given right-hand side and can be used in place of $\underset{\sim}{\rho}$. To this end, we now embark on a small diversion into MILP primal functions.

MILP Primal Functions. In contrast with dual functions, strong *primal functions* bound the value function from above.

Definition 5 (Primal Function). *A primal function $P : \mathbb{R}^{m_2} \to \mathbb{R} \cup \{\pm\infty\}$ is one that satisfies $P(\beta) \geq \phi(\beta)$ for all $\beta \in \mathbb{R}^{m_2}$. It is* strong *at $\hat{\beta} \in \mathbb{R}^{m_2}$ if $P(\hat{\beta}) = \phi(\hat{\beta})$.*

An obvious way to construct such a function is to consider the value function of a restriction of the given MILP (see Güzelsoy [2009] and Güzelsoy and Ralphs [2007] for methods of construction). The following theorem presents the main result for constructing strong primal functions from restrictions of the given MILP.

Theorem 4. *[Güzelsoy, 2009] Consider the MILP value function (MILP-VF). Let $K \subseteq N := \{1, \dots, n_2\}$, $s_i \in \mathbb{R}_+\ \forall i \in K$ be given, and define the function $\bar{\phi} : \mathbb{R}^{m_2} \to \mathbb{R} \cup \{\pm\infty\}$ such that*

$$\bar{\phi}(\beta) = \sum_{i \in K} d_i^2 s_i + \phi_{N \setminus K}(\beta - \sum_{i \in K} G_i^2 s_i) \quad \forall \beta \in \mathbb{R}^{m_2}, \tag{MILP-UBF}$$

where G_i^2 is the i^{th} column of G^2 and

$$\begin{aligned} \phi_{N\setminus K}(\beta) = \quad & \min \sum_{i \in N\setminus K} d_i^2 y_i \\ & s.t. \sum_{i \in N\setminus K} G_i^2 y_i \geq \beta \\ & y_i \in \mathbb{Z}_+\ \forall i \in I,\ y_i \in \mathbb{R}_+\ \forall i \in C, \end{aligned}$$

where $I \subseteq N$ and $C \subseteq N$ represent sets of indices for integer and continuous variables respectively. Then, $\bar{\phi}$ is a valid primal function of ϕ, i.e., $\bar{\phi}(\beta) \geq \phi(\beta)\ \forall \beta \in \mathbb{R}^{m_2}$, if $s_i \in \mathbb{Z}_+\ \forall i \in I \cap K$ and $s_i \in \mathbb{R}_+\ \forall i \in C \cap K$. Further, the function $\bar{\phi}$ will be strong at a given right-hand side if $s_i\ \forall i \in K$ corresponds to the optimal solution values of $y_i\ \forall i \in K$ at this given right-hand side.

The above result indicates that a primal function obtained from a restriction in which the values of certain variables have been fixed is strong at $\hat{\beta} \in \mathbb{R}^{m_2}$ if the fixed values of these variables correspond

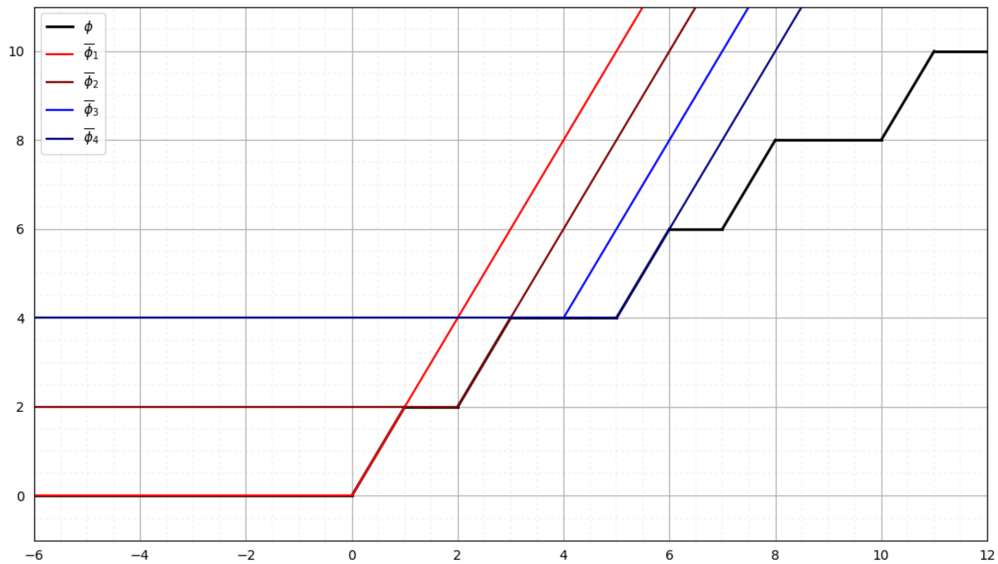


Figure 6: Primal functions for (MILP-Toy)

to those of an optimal solution at $\hat{\beta}$. A convenient restriction is the continuous restriction mentioned earlier, obtained by fixing all integer variables to their optimal values. The resulting value function ϕ_C is nothing but the value function of an LP discussed briefly in Section 3.1. Let us now look at an example of using continuous restrictions to generate primal functions.

Example 5 Consider the MILP (MILP-Toy). Figure 6 demonstrates four primal functions obtained upon applying the result in Theorem 4 to this MILP for right-hand sides $\beta = 0, 2, 4, 5$. Here, we consider continuous restrictions of the given MILP. For example, for $\beta = 0$, the optimal solution of (MILP-Toy) is $(0, 0, 0, 0)$ resulting in the following continuous restriction.

$$\phi_C(\beta) = \min \{4y_4 \mid 2y_4 \geq \beta,\ y_4 \in \mathbb{R}_+\} \quad \forall \beta \in \mathbb{R}$$

It is easy to observe that this LP value function is

$$\phi_C(\beta) = \begin{cases} 0 & \text{if } \beta \leq 0 \\ 2\beta & \text{otherwise} \end{cases},$$

which itself is the required primal function $\bar{\phi}$ at $\beta = 0$ because the integer component of the MILP objective function is zero. Other primal functions can be constructed in a similar way by solving the MILP with a new right-hand side, calculating the integer component of the MILP objective function at its optimal solution, and simply translating ϕ_C based on this integer component value. The primal function can be strengthened in the same way as dual functions are strengthened, by considering the minimum of multiple such (strong) functions. The epigraph of such a function is the minimum of convex radial cones and equals the epigraph of the value function when enough such cones are considered, as mentioned in Section 3.2.1. □

Because we do not expect to be able to obtain a full description of this LP value function, we consider a partial function that we can obtain relatively easily. Specifically, we use a single hyperplane of (LP-VF-Struct) to construct $\bar{\phi}$. In Example 5, this is equivalent to considering only one of the two hyperplanes forming the cone corresponding to the LP value function. It is obvious from Figure 6 that any single hyperplane cannot form a valid primal function in the entire domain of the right-hand side vector. Therefore, we also need to restrict the domain of the right-hand side vector to only the region in which the single hyperplane is a valid upper bound.

Let $y^* = (y^*_I, y^*_C)$ be an optimal solution of an instance of (ReF) for a known right-hand side $(\hat{\beta}^1, \hat{\beta}^2)$, where I and C correspond to sets of indices of second-stage integer and continuous variables respectively. This inherently implies that $\phi(\hat{\beta}^2) = d^2 y^*$. The value function ϕ_C of the continuous restriction is then

$$\begin{aligned} \phi_C(\beta) = \min\ & d^2_C y_C \\ \text{s.t. } & G^2_C y_C \geq \beta \\ & y_C \geq 0. \end{aligned}$$

Let η^* be a dual optimal solution of this LP with right-hand side $\beta^2 - G^2_I y^*_I$. Then $\eta^* \beta$ is a dual function strong at $\beta^2 - G^2_I y^*_I$. From the theory of LP duality, we know that this function provides a valid upper bound as long as η^* remains optimal, which is the case for all β such that $(G^2_B)^{-1}\beta \geq 0$, where B is the index set corresponding to the optimal basis and G^2_B is the optimal basis matrix.

Thus, we obtain our final primal function by restricting the domain to obtain the final form of the function

$$\bar{\phi} = \begin{cases} (\beta^2 - G^2_I y^*_I)\eta^* + d^2_I y^*_I & \text{if } (G^2_B)^{-1}(\beta^2 - G^2_I y^*_I) \geq 0 \\ \infty & \text{otherwise} \end{cases} \qquad \text{(MILP-RPUBF)}$$

with which we replace ϕ in (ReF-LBF). This ensures that the dual function $\underline{\rho}$ that we construct in each iteration of the algorithm is valid for all values of (β^1, β^2).

4.3 Master Problem

Combining the results obtained in the previous section, the final form of $\underline{\rho}$ is

$$\underline{\rho}(\beta^1, \beta^2) = \min_{t \in T} \left(\beta^1 \eta^{1,t} + \beta^2 \eta^{2,t} + \bar{\phi} \eta^{\phi,t} + \alpha^t \right), \qquad \text{(ReF-LBF-Final)}$$

where $\bar{\phi}$ is the primal function (MILP-RPUBF). This results in the optimality constraint

$$z \geq \min_{t \in T} \left(\beta^1 \eta^{1,t} + \beta^2 \eta^{2,t} + \bar{\phi} \eta^{\phi,t} + \alpha^t \right) \qquad \text{(MIBLP-OC)}$$

that we add to the master problem in each iteration of the algorithm, with $(\beta^1, \beta^2) = (b^1 - A^1x, b^2 - A^2x)$. Finally, the updated master problem after iteration k of the algorithm is

$$\begin{aligned}
&\min cx + z \\
&\text{s.t. } z \geq \min_{t \in T_i} \{(b^1 - A^1x)\eta_i^{1,t} + (b^2 - A^2x)\eta_i^{2,t} + \bar{\phi}_i \eta_i^{\phi,t} + \alpha_i^t\} \quad 1 \leq i \leq k \\
&\quad \bar{\phi}_i = \begin{cases} (b^2 - A^2x - G_I^2 y_{I,i}^*)\eta_i^* + d_I^2 y_{I,i}^* & \text{if } (G_{B,i}^2)^{-1}(b^2 - A^2x - G_I^2 y_{I,i}^*) \geq 0 \\ \infty & \text{otherwise} \end{cases} \quad 1 \leq i \leq k \\
&\quad x \in X,
\end{aligned} \tag{MIBLP-MP-Updated}$$

where the vectors, matrices and sets with the subscript i correspond to the information obtained in iteration $i \leq k$ of the algorithm.

4.4 Overall Algorithm

We now have all the components required for solving (MIBLP) with the generalized Benders' decomposition algorithm in Figure 1. In each iteration k of the algorithm, a master problem of the form (MIBLP-MP-Updated) is solved to obtain its optimal solution (x^k, z^k) and a global lower bound. Then, the subproblem is solved as an equivalent MILP, by evaluating (ReF) at $(\beta^1, \beta^2) = (b^1 - A^1x^k, b^2 - A^2x^k)$, to obtain a branch-and-bound tree and a global upper bound. Then, an optimality constraint of the form (MIBLP-OC) is constructed and added to the master problem to strengthen z. This constraint introduces some nonlinear components in the master problem but they can be linearized (mentioned below) to obtain an MILP formulation for the master problem. These steps are repeated until the termination criterion is achieved.

We now illustrate the above discussion with an example.

Example 6 Consider the MIBLP

$$\begin{aligned}
\min\ & x_1 - 3x_2 - y_1 + y_2 - 5y_3 + y_4 \\
\text{s.t. } & -x_1 + 2x_2 \leq 1 \\
& x_1 \leq 3,\ x_2 \leq 2,\ x_1, x_2 \in \mathbb{Z}_+ \\
(y_1, y_2, y_3, y_4) \in\ & \arg\min \{2\check{y}_1 + 4\check{y}_2 + 3\check{y}_3 + 4\check{y}_4 \\
& \quad \text{s.t. } 2\check{y}_1 + 5\check{y}_2 + 2\check{y}_3 + 2\check{y}_4 \geq x_1 + x_2 \\
& \quad \check{y}_1, \check{y}_2, \check{y}_3, \in \mathbb{Z}_+,\ \check{y}_4 \in \mathbb{R}_+\},
\end{aligned} \tag{MIBLP-Toy}$$

which is based on (MILP-Toy) and (ReF-Toy). Based on earlier discussion, we solve four optimization problems in iteration k of the algorithm: a master problem, an MILP (MILP-Toy) (with $\beta^k = x_1^k + x_2^k$), a subproblem (ReF-Toy) (with $\beta^k = x_1^k + x_2^k$), and a continuous restriction of (MILP-Toy).

Iteration 1. Our initial dual function is simply $\underline{\rho}^0(\beta) = -\infty$ for all $\beta \in \mathbb{R}^{m_2}$ and solving the initial master problem yields the optimal solution $(x_1^1, x_2^1) = (3, 2)$ and $z^1 = -\infty$, so that $\text{LB}^1 = -\infty$. Then, we solve (MILP-Toy) with right-hand side $x_1^1 + x_2^1 = 5$ to obtain $\phi(x_1^1 + x_2^1) = 4$. Next, we solve the subproblem to obtain its optimal solution $(y_1^1, y_2^1, y_3^1, y_4^1) = (0, 1, 0, 0)$, so we have

t	$(\eta_1^{2,t}, \eta_1^{\phi,t})$	$(\underline{\eta}_1^{1,t},\ \underline{\eta}_1^{2,t}, \underline{\eta}_1^{3,t}, \underline{\eta}_1^{4,t})$	$(\bar{\eta}_1^{1,t}, \bar{\eta}_1^{2,t}, \bar{\eta}_1^{3,t}, \bar{\eta}_1^{4,t})$	$\eta_1^{2,t}\beta + \eta_1^{\phi,t}\bar{\phi}_1(\beta) + \alpha_1^t$
1	(3.29, -3.86)	$(0.14, 0, 0, 9.86)$	$(0, 0, 0, 0)$	$3.29\beta - 3.86\ \bar{\phi}_1(\beta)$

Table 4: Data for construction of the dual function in Example 6.

$\rho(x_1^1 + x_2^1) = 1$ and $\text{UB}^1 = x_1^1 - 3x_2^1 + \rho(x_1^1 + x_2^1) = -2$. We obtain the dual information (dual solution, positive and negative reduced costs) shown in Table 4 from the branch-and-bound tree, which has only one node. Since $\text{UB}^1 \neq \text{LB}^1$, we further solve the continuous restriction to obtain its optimal dual solution $\eta_1^* = 0$ and optimal basis inverse matrix $(G_{B,1}^2)^{-1} = [-1]$. Finally, we construct and add the dual function

$$\underline{\rho}^1(\beta) = \min\{3.29\beta - 3.86\bar{\phi}_1(\beta)\} = 3.29\ \beta - 3.86\bar{\phi}_1(\beta),$$

where

$$\bar{\phi}_1(\beta) = \begin{cases} 4 & \text{if } \beta \leq 5 \\ \infty & \text{otherwise.} \end{cases},$$

to the master problem and proceed to the next iteration.

For conciseness, we now mention only $\underline{\rho}^k(\beta)$ and $\bar{\phi}_k(\beta)$ obtained in each iteration k.

Iteration 2.

$$\underline{\rho}^2(\beta) = -\frac{5}{3}\bar{\phi}_2(\beta)$$

$$\bar{\phi}_2(\beta) = \begin{cases} 0 & \text{if } \beta \leq 0 \\ \infty & \text{otherwise} \end{cases}$$

Iteration 3.

$$\underline{\rho}^3(\beta) = \min\left\{0.5\bar{\phi}_3(\beta), -4.5\ \bar{\phi}_3(\beta) + 8.5\right\}$$

$$\bar{\phi}_3(\beta) = \begin{cases} 2 & \text{if } \beta \leq 2 \\ \infty & \text{otherwise} \end{cases}$$

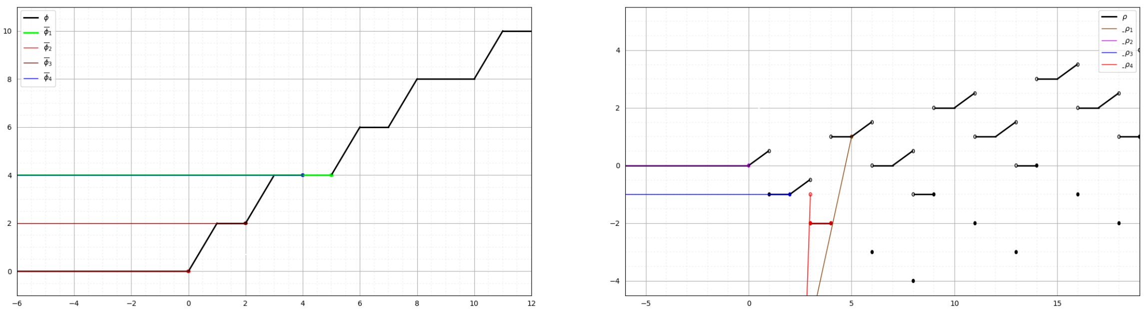
Iteration 4.

$$\underline{\rho}^4(\beta) = \min\left\{-\frac{5}{3}\bar{\phi}_4(\beta) + \ \frac{23}{3}, 17\beta - 13\bar{\phi}_4(\beta),\ -0.5\bar{\phi}_4(\beta),\ 21\beta - 26\bar{\phi}_4(\beta) + 40\right\}$$

$$\bar{\phi}_4(\beta) = \begin{cases} 4 & \text{if } \beta \leq 4 \\ \infty & \text{otherwise} \end{cases}$$

Iteration 5. Solving the updated master problem yields $(x_1^5, x_2^5, z^5) = (1, 1, -1)$ and $\text{LB}^5 = -3$. Solving the subproblem yields $(y_1^5, y_2^5, y_3^5, y_4^5) = (1, 0, 0, 0)$, $\rho(x_1^5 + x_2^5) = -1$ and $\text{UB}^5 = -3$. Since $\text{UB}^5 = \text{LB}^5$, the termination criterion is achieved. Hence, we stop the algorithm with an optimal solution $(x_1^*, x_2^*, y_1^*, y_2^*, y_3^*, y_4^*) = (1, 1, 1, 0, 0, 0)$.

Figure 7a shows the value function ϕ and its primal functions obtained in every iteration. Similarly, Figure 7b shows the reaction function ρ and its dual functions. The function values are infinite



(a) MILP Primal Functions (b) Reaction Function Dual Functions

Figure 7: Functions constructed in the algorithm for solving (MIBLP-Toy)

wherever there is no plot. These figures illustrate the fact that overall approximations of ϕ and ρ are strengthened after every iteration. After the final iteration, the approximation function values are the same as the exact function values at the right-hand side $x_1^* + x_2^* = 2$. □

To conclude, we briefly discuss the linearization of the master problem (MIBLP-MP-Updated). For notational simplicity, we drop the subscript denoting the iteration. There are two types of nonlinearities in this problem: (1) the if-else condition in (MILP-RPUBF) and (2) the minimization operator in (ReF-LBF-Final). We eliminate these nonlinearities by introducing binary variables and big-M parameters. This results in the following MILP form of the master problem.

$$\min cx + z$$
$$\text{s.t. } z \geq (b^1 - A^1x)\eta^{1,t} + (b^2 - A^2x)\eta^{2,t} + \bar{\phi}\eta^{\phi,t} + \alpha^t - M_D(1 - u_t) \tag{c1}$$
$$\sum_{t \in T} u_t = 1 \tag{c2}$$
$$\bar{\phi} = (b^2 - A^2x - G_I^2 y_I^*)\eta^* + d_I^2 y_I^* + M_P v \tag{c3}$$
$$-\underline{M}v^1 \leq (G_B^2)^{-1}(b^2 - A^2x - G_I^2 y_I^*) \leq \bar{M}(1 - v^1) - \epsilon \tag{c4}$$
$$Mv - \sum_{i \in \{1,\dots,m_2\}} v_i^1 \geq 0 \tag{c5}$$
$$v - \sum_{i \in \{1,\dots,m_2\}} v_i^1 \leq 0 \tag{c6}$$
$$x \in X, z \text{ free}, \bar{\phi} \text{ free}, u_t \in \mathbb{B}\ \forall t \in T, v \in \mathbb{B}, v_i^1 \in \mathbb{B}\ \forall i \in \{1, \dots, m_2\}$$

Constraints (c1)-(c2) eliminate the minimization operator by adding the binary variables u_t for $t \in T$ and the big-M parameter M_D. Constraints (c3)-(c6) eliminate the if-else condition by adding the binary variables v, v_i^1 for $i \in \{1, \dots, m_2\}$ and the big-M parameters $\underline{M}$, $\bar{M}$ and M. Specifically, the constraints (c4)-(c6) impose domain restriction on the first-stage variables w.r.t. the primal function (MILP-RPUBF). This in turn restricts the domain of the right-hand sides for which ϕ may be evaluated. The main idea is to set some $v_i^1 = 0$ whenever $(G_{B_i}^2)^{-1}(b^2 - A^2x - G_I^2 y_I^*) \geq 0$

and to set $v_i^1 = 1$ whenever $(G^2_{B_i})^{-1}(b^2 - A^2x - G^2_I y^*_I) < 0$, where $(G^2_{B_i})^{-1}$ is the i^{th} row of the optimal basis matrix inverse. If at least one $v_i^1 = 1$, then $v = 1$ further implying $\bar{\phi}$ will have a very large value which is as required. If all $v_i^1 = 0$, then $v = 0$ further implying $\bar{\phi}$ will have a finite value which is also as required. Finally, ϵ is also a parameter corresponding to the domain restriction constraints added to deal with the strict inequality arising from "otherwise" condition in (MILP-RPUBF), and is the trickiest of all parameters to evaluate.

5 Multistage Mixed Integer Linear Optimization Problems

In this final section, we discuss the application of the generalized Benders' principle to the general class of MMILPs that is the natural extension of MIBLPs to l stages. Just like MIBLPs, an l-stage MMILP can be formulated as a standard mathematical optimization problem by considering a constraint requiring values of all but first-stage variables to be optimal for an l-1-stage MMILP that is parametric in the first-stage variables, in addition to the usual linear constraints. Then, assuming that all input vectors and matrices are rational of appropriate dimensions without loss of generality, we have an l-stage MMILP with a parametric right-hand side β defined as

$$\begin{aligned} \text{MMILP}^l(\beta) = \min\ & d^{11}x^1 + d^{12}x^2 + \ldots + d^{1l}x^l \\ \text{s.t. }\ & A^{11}x^1 + A^{12}x^2 + \ldots + A^{1l}x^l \geq \beta \\ & x^1 \in X^1 \\ & (x^2, x^3, \ldots, x^l) \in \text{optimal set of } \text{MMILP}^{l-1}(b^2 - A^{21}x^1), \end{aligned} \tag{MMILP^l}$$

where $\text{MMILP}^{l-1}(b^2 - A^{21}x^1)$ denotes an $l-1$-stage MMILP with the parametric right-hand side $b^2 - A^{21}x^1$, which in turn is a linear function of first-stage variables, defined similar to (MMILP^l) but with $l-1$ variable vectors, as

$$\begin{aligned} \text{MMILP}^{l-1}(\beta) = \min\ & d^{22}x^2 + d^{23}x^3 + \ldots + d^{2l}x^l \\ \text{s.t. }\ & A^{22}x^2 + A^{23}x^3 + \ldots + A^{2l}x^l \geq \beta \\ & x^2 \in X^2 \\ & (x^3, x^4, \ldots, x^l) \in \text{optimal set of } \text{MMILP}^{l-2}(b^3 - A^{31}x^1 - A^{32}x^2). \end{aligned} \tag{MMILP^{l-1}}$$

These formulations exhibit the natural recursive property of MMILPs that we spoke about in the beginning of the paper. It should be clear by now why this recursive structure also means that the Benders' framework makes it easy to envision algorithms for solving such problems (whether these algorithms are practical is another question).

We thus apply the operations of generalized Benders' decomposition framework here. Upon projecting (MMILP^l) (for a fixed $\beta = b^1$) into the space of the first-stage variables, we obtain the reformulation

$$\min \left\{ d^{11}x^1 + \rho^{l-1}(b^1 - A^{11}x^1,\ b^2 - A^{21}x^1) \mid x^1 \in X^1 \right\}, \tag{MMILP^l-ReF}$$

where ρ^{l-1} is an l-1-stage reaction function analogous to ρ in (ReF), defined as

$$\begin{aligned}\rho^{l-1}(\beta^1, \beta^2) = \min\ & d^{12}x^2 + d^{13}x^3 + \ldots + d^{1l}x^l \\ \text{s.t. }\ & A^{12}x^2 + A^{13}x^3 + \ldots + A^{1l}x^l \geq \beta^1 \\ & (x^2, x^3, \ldots, x^l) \in \text{optimal set of MMILP}^{l-1}(\beta^2).\end{aligned} \tag{ReF^{l-1}}$$

This results in the master problem

$$\begin{aligned}\min\ & d^{11}x^1 + z \\ \text{s.t. }\ & z \geq \underline{\rho}^{l-1}(b^1 - A^{11}x^1,\ b^2 - A^{21}x^1) \\ & x^1 \in X^1,\end{aligned} \tag{MMILP^l-MP}$$

where $\underline{\rho}^{l-1}$ is a (strong) dual function constructed by solving the subproblem, which is nothing but evaluating (ReF^{l-1}) at a known right-hand side.

This subproblem itself involves an l-1-stage MMILP (MMILP^{l-1}), and solution of it calls for solving this l-1-stage MMILP. But this l-1-stage MMILP can also be solved with the generalized Benders' principle due to the recursive structure. For constructing (strong) dual functions, recall that the subproblem in the context of MIBLPs is solved as an equivalent MILP. This allowed us to construct (ReF-LBF-Final) with the help of Theorem 3. A similar approach can be adopted here for evaluating the $l-1$-stage reaction function by reformulating it using an l-2-stage function. This allows us to construct required (strong) dual functions in the current context similar to (ReF-LBF-Final).

While this approach obviously results in an algorithm whose running time will be super-exponential, it nevertheless provides the intuition as to exactly *why* these problems are so hard to solve and what the polynomial time hierarchy is really all about.

6 Conclusions

We have described a generalization of Benders' decomposition framework and illustrated its principles by applying it to several well-known classes of optimization problems that fall under the broad umbrella of MMILPs. The development of the abstract framework and its application to the class of MIBLPs is our main contribution. These stemmed from our observation that Benders' framework can be viewed as a procedure for iterative refinement of dual functions associated with the value function of the second-stage problem and that this basic concept can be applied to a wide range of problems defined by additively separable functions. These observations resulted in the development of the generalized Benders' decomposition algorithm for solving MIBLPs, as well as a conceptual extension to the case of general MMILPs. Although not discussed here, this framework can be readily applied to even broader classes of problems, such as those discussed in Bolusani et al. [2020], which incorporate stochasticity. While the algorithms could be of practical interest, the algorithmic abstraction itself serves to illustrate basic theoretical principles, such as concepts of general duality and why k-stage MMILPs are canonical hard problems for stage k of the polynomial time hierarchy.

The algorithms described here are naive in the sense that their efficient implementations for practical purposes would require substantial additional development, especially for the classes of MIBLPs and MMILPs. To this end, our plans include enhancement of these algorithms by working in the areas of preprocessing techniques, warm starting of master and subproblem solves, cut management, a branch-and-Benders' framework, alternative linearization techniques, and other enhancements.

References

A. Bachem and R. Schrader. Minimal equalities and subadditive duality. *Siam J. on Control and Optimization*, 18(4):437–443, 1980.

B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer. *Non-linear parametric optimization.* Birkhäuser verlag, 1983.

J. Bard and J.T. Moore. An algorithm for the discrete bilevel programming problem. *Naval Research Logistics*, 39(3):419–435, 1992.

L. Baringo and A.J. Conejo. Transmission and wind power investment. *IEEE Transactions on Power Systems*, 27(2):885–893, May 2012.

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

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

C.E. Blair. A closed-form representation of mixed-integer program value functions. *Mathematical Programming*, 71(2):127–136, 1995.

C.E. Blair and R.G. Jeroslow. The value function of a mixed integer program: I. *Discrete Mathematics*, 19(2):121–138, 1977.

C.E. Blair and R.G. Jeroslow. The value function of a mixed integer program: Ii. *Discrete Mathematics*, 25(1):7–19, 1979a.

C.E. Blair and R.G. Jeroslow. The value function of a mixed integer program: II. *Discrete Mathematics*, 25:7–19, 1979b.

C.E. Blair and R.G. Jeroslow. The value function of an integer program. *Mathematical Programming*, 23(1):237–273, 1982. ISSN 0025-5610.

C.E. Blair and R.G. Jeroslow. Constructive characterizations of the value-function of a mixed-integer program i. *Discrete Applied Mathematics*, 9(3):217–233, 1984.

S. Bolusani, S. Coniglio, T.K. Ralphs, and S. Tahernejad. A Unified Framework for Multistage Mixed Integer Linear Optimization. Technical report, COR@L Laboratory Technical Report 20T-005, Lehigh University, 2020. URL http://coral.ie.lehigh.edu/~ted/files/papers/MultistageFramework20.pdf.

J. Bracken and J.T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973.

A. Caprara, M. Carvalho, A. Lodi, and G. Woeginger. Bilevel knapsack with interdiction constraints. *INFORMS Journal on Computing*, 28(2):319–333, 2016.

M. Caramia and R. Mari. Enhanced exact algorithms for discrete bilevel linear problems. *Optimization Letters*, 9(7):1447–1468, 2015.

S. DeNegre. *Interdiction and Discrete Bilevel Linear Programming*. PhD, Lehigh University, 2011. URL http://coral.ie.lehigh.edu/~ted/files/papers/ScottDeNegreDissertation11.pdf.

S. DeNegre and T.K. Ralphs. A Branch-and-Cut Algorithm for Bilevel Integer Programming. In *Proceedings of the Eleventh INFORMS Computing Society Meeting*, pages 65–78, 2009. doi: 10.1007/978-0-387-88843-9_4. URL http://coral.ie.lehigh.edu/~ted/files/papers/BILEVEL08.pdf.

N.P. Faísca, V. Dua, B. Rustem, P.M. Saraiva, and E.N. Pistikopoulos. Parametric global optimisation for bilevel programming. *Journal of Global Optimization*, 38:609–623, 2007.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172:77–103, 2018.

L.P. Garcés, A.J. Conejo, R. García-Bertrand, and R. Romero. A bilevel approach to transmission expansion planning within a market environment. *IEEE Transactions on Power Systems*, 24(3): 1513–1522, Aug 2009.

A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, Oct 1972. ISSN 1573-2878. doi: 10.1007/BF00934810. URL https://doi.org/10.1007/BF00934810.

M. Güzelsoy. *Dual Methods in Mixed Integer Linear Programming*. PhD, Lehigh University, 2009. URL http://coral.ie.lehigh.edu/~ted/files/papers/MenalGuzelsoyDissertation09.pdf.

M. Güzelsoy and T.K. Ralphs. Duality for Mixed-Integer Linear Programs. *International Journal of Operations Research*, 4:118–137, 2007. URL http://coral.ie.lehigh.edu/~ted/files/papers/MILPD06.pdf.

A. Hassanzadeh. *Two-Stage Stochastic Mixed Integer Optimization*. PhD, Lehigh, 2015.

A. Hassanzadeh and T.K. Ralphs. A Generalized Benders' Algorithm for Two-Stage Stochastic Program with Mixed Integer Recourse. Technical report, COR@L Laboratory Technical Report 14T-005, Lehigh University, 2014a. URL http://coral.ie.lehigh.edu/~ted/files/papers/SMILPGenBenders14.pdf.

A. Hassanzadeh and T.K. Ralphs. On the Value Function of a Mixed Integer Linear Optimization Problem and an Algorithm for Its Construction. Technical report, COR@L Laboratory Technical Report 14T-004, Lehigh University, 2014b. URL http://coral.ie.lehigh.edu/~ted/files/papers/MILPValueFunction14.pdf.

M. Hemmati and J.C. Smith. A mixed integer bilevel programming approach for a competitive set covering problem. Technical report, Clemson University, 2016.

J.N. Hooker and G. Ottosson. Logic-based benders decomposition. *Mathematical Programming*, 96 (1):33–60, Apr 2003.

R.G. Jeroslow. Cutting plane theory: Algebraic methods. *Discrete Mathematics*, 23:121–150, 1978.

R.G. Jeroslow. Minimal inequalities. *Mathematical Programming*, 17:1–15, 1979.

E.L. Johnson. Cyclic groups, cutting planes, and shortest paths. In T.C. Hu and S.M. Robinson, editors, *Mathematical Programming*, pages 185–211. Academic Press, New York, NY, 1973.

E.L. Johnson. On the group problem for mixed integer programming. *Mathematical Programming Study*, 2:137–179, 1974.

M. Köppe, M. Queyranne, and C. T. Ryan. Parametric integer programming algorithm for bilevel mixed integer programs. *Journal of Optimization Theory and Applications*, 146(1):137–150, Jul 2010.

L. Lozano and J.C. Smith. A value-function-based exact approach for the bilevel mixed-integer programming problem. *Operations Research*, 65(3):768–786, 2017.

J.T. Moore and J.F. Bard. The mixed integer linear bilevel programming problem. *Operations research*, 38(5):911–921, 1990.

George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., 1988.

G.K. Saharidis and M.G. Ierapetritou. Resolution method for mixed integer bi-level linear problems based on decomposition technique. *Journal of Global Optimization*, 44(1):29–51, 2008.

N.V. Sahinidis and I.E. Grossmann. Convergence properties of generalized benders decomposition. *Computers & Chemical Engineering*, 15(7):481 – 491, 1991. ISSN 0098-1354. doi: https://doi.org/10.1016/0098-1354(91)85027-R. URL http://www.sciencedirect.com/science/article/pii/009813549185027R.

A. Shapiro. Monte carlo sampling methods. *Handbooks in operations research and management science*, 10:353–425, 2003.

S. Tahernejad, T.K. Ralphs, and S.T. DeNegre. A Branch-and-Cut Algorithm for Mixed Integer Bilevel Linear Optimization Problems and Its Implementation. *Mathematical Programming Computation (to appear)*, 2016. URL http://coral.ie.lehigh.edu/~ted/files/papers/MIBLP16.pdf. To appear, Mathematical Programming Computation.

R.M. Van Slyke and R. Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, pages 638–663, 1969.

L. Vicente, G. Savard, and J. Júdice. Discrete linear bilevel programming problem. *Journal of Optimization Theory and Applications*, 89(3):597–614, 1996.

Heinrich Von Stackelberg. *Marktform und Gleichgewicht*. Julius Springer, 1934.

L. Wang and P. Xu. The watermelon algorithm for the bilevel integer linear programming problem. *SIAM Journal on Optimization*, 27(3):1403–1430, 2017.

U.P. Wen and Y.H. Yang. Algorithms for solving the mixed integer two-level linear programming problem. *Computers & Operations Research*, 17(2):133–142, 1990.

R. Wollmer. Removing arcs from a network. *Operations Research*, 12(6):934–940, 1964.

L.A. Wolsey. Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming*, 20(1):173–195, 1981a. ISSN 0025-5610.

L.A. Wolsey. Integer programming duality: Price functions and sensitivity analaysis. *Mathematical Programming*, 20:173–195, 1981b.

P. Xu and L. Wang. An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions. *Computers & operations research*, 41:309–318, 2014.

Bo Zeng and Yu An. Solving bilevel mixed integer program by reformulations and decomposition. Technical report, University of South Florida, 2014. URL http://www.optimization-online.org/DB_FILE/2014/07/4455.pdf.