DUAL METHODS IN MIXED INTEGER LINEAR PROGRAMMING

by

Menal Guzelsoy

Presented to the Graduate and Research Committee
of Lehigh University
in Candidacy for the Degree of
Doctor of Philosophy

in

Industrial and Systems Engineering

Lehigh University

January 2010

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Date

Dr. Theodore K. Ralphs
Dissertation Advisor

Accepted Date

Committee:

Dr. Theodore K. Ralphs, Chairman

Dr. Sanjeeb Dash

Dr. Garth Isaak

Dr. Jeffrey T. Linderoth

Acknowledgments

Foremost, I would like to express my deep and sincere gratitude to my advisor, Ted Ralphs, for his guidance, patience, and continuous support throughout this study. He has been an excellent advisor with his immense knowledge and constructive comments; a great teacher with his logical way of thinking, motivation and enthusiasm; and a good friend with his warm and positive attitude.

I would also like to thank my thesis committee Sanjeeb Dash, Garth Isaak and Jeff Linderoth for their time to provide sound advice and valuable criticism for completing this thesis.

I am grateful to my friends and fellow colleagues Ashutosh Mahajan, Scott DeNegre, Serdar Ozdemir, Gulistan Mese, Zumbul Bulut Atan, Oguz Atan, Berrin Aytac, Burak Bekisli, Figen Akdin Bekisli, Cagri Latifoglu, Ruken Duzgun, Tolga Seyhan, Svetlana Oshkai and Yan Xu for their support and encouragement during this journey. Special thanks go to Zeliha Akca, Banu Gemici-Ozkan, Umit Ozkan, Gokhan Metan and Demet Metan for standing beside me along this path and for all the time, joy and sorrow we shared together.

Finally, I would like to extend my heartfelt gratitude to my family, Aynur, Ismet, Esra, Servin and Unal. This thesis would not have been possible without their unbounded love and unconditional support.

Contents

Abstract

The primary goal of this thesis is to investigate the extent to which the dual methods available for linear programs (LPs) can be extended to the case of mixed integer linear programs (MILPs) in which a specified subset of the variables of an LP must take integer values.

A central focus of duality in the integer programming setting is the *value function* that returns the optimal solution value of an MILP as a function of the right-hand side. Because it is difficult to obtain the value function explicitly, we focus on methods for obtaining approximations of the value function. We first discuss methods for constructing *dual functions*, which bound the value function from below, and discuss how these can be obtained from primal solution algorithms including branch-and-cut, the most common method in practice for solving MILPs.

Next, we detail the results of our theoretical examination of the structure of the value function, focusing on the case of an MILP with a single constraint. We show that the value function in this case is uniquely determined by a finite number of break points and is a piecewise-linear function with at most two slopes. We derive conditions under which the value function is continuous and suggest a method for systematically extending the value function from a specified neighborhood of the origin to the entire real line. Although we focus first on the case of an MILP with a single constraint for illustration, we extend a number of these results to more general settings.

For the value function of a general MILP, we discuss approximation methods that leverage our knowledge of the value functions of single row relaxations and other dual functions. We outline methods that enable us to use these approximations as a substitute for the value function in order to solve large instances of certain classes of multi-stage mathematical programming. We illustrate these methods for both stochastic mixed integer programs and mixed integer bilevel programs.

Finally, we discuss how dual information obtained from primal solution algorithms can be used for sensitivity analysis and warm starting the solution procedure of a modified problem from an advanced level. We give computational results for various applications including iterative combinatorial auctions, capacitated vehicle routing problems, feasibility algorithms (RINS), stochastic integer and bicriteria integer programs.

Chapter 1

Introduction

Consider a general *mathematical programming* problem

$$Z_P = \min_{x \in U} \quad f(x) \,, \tag{1.1}$$

where $f : \Upsilon \rightarrow \mathbb{R}$ is an arbitrary function and $U \subseteq \Upsilon$ is the feasible region. We call (1.1) the *primal problem* and any vector $x \in U$ a *feasible solution*. We let $Z_P = -\infty$ if for any $x \in U$, there exists $y \in U$ such that $f(y) < f(x)$, and by convention, $Z_P = \infty$ if $U = \emptyset$. The goal is to determine Z_P along with an *optimal solution* x^* such that $f(x^*) = Z_P$, if Z_P is finite, where $f(x)$ is the *solution value* of x.

In many applications, we are interested in a parameterized version of the primal problem. Let $\mathscr{X}$ be a collection of subsets of Υ and consider the function $Z : \mathscr{X} \rightarrow \mathbb{R} \cup \{\pm\infty\}$ defined as

$$Z(X) = \min_{x \in X} \quad f(x) \tag{1.2}$$

for all $X \in \mathscr{X}$. Here, $\mathscr{X}$ is a collection of feasible regions defining a family of optimization problems and Z is the corresponding *value function*. Then, a *dual function* $F : \mathscr{X} \rightarrow \mathbb{R} \cup \{\pm\infty\}$ is a function that bounds the value function over the set $\mathscr{X}$, that is,

$$F(X) \leq Z(X) \quad \forall X \in \mathscr{X}. \tag{1.3}$$

Duality in mathematical programming is the study of such dual functions and the methods for constructing them. The aim in constructing a dual function F is for it to closely approximate the value function Z. Clearly, the best dual function in this respect is the value function itself. In some cases, the structure of the set $\mathscr{X}$ and the objective function f might be such that the value function can be explicitly constructed. However, this has proven difficult in general and the question therefore arises exactly how to select such a function from among the possible alternatives. The most frequently used method is to choose a dual function that would yield the best bound for a given base instance of particular interest denoted by $\bar{X}$, by solving a *dual problem*

$$Z_D = \max\{F(\bar{X}) \mid F(X) \leq Z(X) \quad \forall X \in \mathscr{X} \ , \ F : \mathscr{X} \rightarrow \mathbb{R} \cup \{\pm\infty\}\} \ , \tag{1.4}$$

which is an optimization problem over the space of dual functions. Among those, we call F a feasible solution, or *a weak dual function*, if it satisfies (1.3), and an optimal solution, or *a strong dual function* with respect to $\bar{X} \in \mathscr{X}$, if it further satisfies $F(\bar{X}) = Z(\bar{X})$. Observe that $Z_D \leq Z_P$ by construction, and if Z_P is finite, then there always exists an optimal strong dual function for (1.4), i.e., $Z_D = Z_P$, since the value function is itself feasible.

1.1 Motivation

An understanding of the structure of the value function is at the core of many optimization tools, such as those for sensitivity analysis and the solution of parametric and miltilevel programs. If the value function is known explicitly, then efficient techniques utilizing the value function can be introduced. In general, it is far from being computationally tractable in practice to obtain the value function explicitly. Instead, so-called dual methods are adopted in which we substitute an approximation of the value function, often a dual function.

For some problem classes, this dual approach yields extremely well-structured and manageable dual methods. Perhaps, the best-studied of these is *linear programs* (LPs), where the above notion of duality has been successful in contributing to both theory and practice. By taking advantage of LP duality, it has been possible to develop not only direct primal-dual solution algorithms for solving LPs but also sophisticated dynamic methods appropriate for sensitivity analysis techniques, methods for re-solving a modified instance from an advanced starting point and methods for solution of large-scale instances. Such procedures are useful in cases in which the input data are subject to fluctuation after the solution procedure has been initiated, in applications for which the solution of a series of closely-related instances is required and in optimization problems where there is a need for the value function of an LP. In particular, these methods are in the core of a variety of optimization algorithms including decomposition algorithms, parametric and stochastic programming algorithms, multi-objective optimization algorithms, and algorithms for analyzing infeasible mathematical models.

Although several efforts have been made to replicate for *mixed integer linear programs* (MILPs) the same duality relation that exists for LPs, it has so far been difficult to derive methods for MILPs that are as functional and efficient as in the LP case. However, with recent advances in computing, it may now be time again to devote increased attention to MILP duality theory in the hope that new computational tools can contribute to more successful implementations.

1.2 Definitions and Notation

A *linear program* is the problem of minimizing a linear objective function over a polyhedral feasible region

$$\mathcal{P} = \{x \in \mathbb{R}^n_+ \mid Ax = b\} \tag{1.5}$$

defined by rational constraint matrix $A \in \mathbb{Q}^{m\times n}$ and right-hand side vector $b \in \mathbb{R}^m$, that is,

$$Z_{LP} = \min_{x\in\mathcal{P}} \quad cx \tag{1.6}$$

for $c \in \mathbb{R}^n$. A *mixed integer linear program* is an LP with the additional constraint that a specified subset of the variables must take on integer values. For the remainder of the thesis, we address the canonical MILP instance specified by (1.5), with the *integer variables* (those required to take on integer values) indexed by the set $I = \{1, \ldots, r\} \subseteq N = \{1, \ldots, n\}$ if $r > 0$ (otherwise, $I = \emptyset$).

The remaining variables, indexed by the set $C = N\backslash I$, constitute the *continuous variables*. The feasible region is then given by $\mathcal{S} = \mathcal{P} \cap (\mathbb{Z}^r \times \mathbb{R}^{n-r})$ and the MILP primal problem can be stated as that of determining

$$Z_{IP} = \min_{x \in \mathcal{S}} cx \; . \tag{1.7}$$

The LP obtained from a given MILP by removing the integrality requirements on the variables, i.e., setting $I = \emptyset$, is referred to as the associated *LP relaxation*. The associated *pure integer linear program* (PILP), on the other hand, is obtained by requiring *all* variables to be integer, i.e., setting $r = n$.

In what follows, we frequently refer to certain classes of functions, defined below.

Definition 1.1 *Let a function f be defined over domain V. Then f is*

- subadditive *if* $f(v_1) + f(v_2) \geq f(v_1 + v_2) \; \forall \, v_1, v_2, v_1 + v_2 \in V$.
- linear *if V is closed under addition and scalar multiplication and*

  i. $f(v_1) + f(v_2) = f(v_1 + v_2) \; \forall \, v_1, v_2 \in V$,

  ii. $f(\lambda v) = \lambda f(v) \; \forall \, v \in V, \; \forall \, \lambda \in \mathbb{R}$.
- convex *if V and epi$(f) = \{(v, y) : v \in V, y \geq f(v)\}$ are convex sets, and*
- polyhedral *if epi(f) is a polyhedron.*

Definition 1.2 *For a given $k \in \mathbb{N}$, let*

- $\Lambda^k = \{f \mid f : \mathbb{R}^k \rightarrow \mathbb{R}\}$,
- $\mathcal{L}^k = \{f \in \Lambda^k \mid f \text{ is linear}\}$,
- $\mathcal{C}^k = \{f \in \Lambda^k \mid f \text{ is convex}\}$,
- $\mathcal{F}^k = \{f \in \Lambda^k \mid f \text{ is subadditive}\}$.

For any index set $K \subseteq N$, A_K is the submatrix consisting of the corresponding columns of A and similarly, y_K is the vector consisting of just the corresponding components of a vector y. The notation $\lceil\lambda\rceil$ for a scalar λ is used to denote the smallest integer greater than or equal to λ. Similarly, we let $\lfloor\lambda\rfloor = -\lceil -\lambda\rceil$. For a function $f \in \Lambda^k$, $\lceil f\rceil$ is the function defined by $\lceil f\rceil(d) = \lceil f(d)\rceil \ \forall d \in \mathbb{R}^k$. Finally, the l_1 norm of a vector $x = (x_1, \ldots, x_n)$ is denoted by $\|x\|_1 = \sum_{i=1}^n |x_i|$.

1.3 Background

Below, we first give an overview of LP duality to illustrate how the dual formulation (1.4) enables practical techniques for optimality conditions, sensitivity analysis, warm-starting, and approximation of the LP value function. Then, we briefly introduce MILP duality and discuss the difficulties of extending LP results to the MILP case.

1.3.1 Linear Programming

Because the right-hand side can be thought of as describing the level of resources available within the system being optimized, it is natural to consider the question of how the optimal solution value of an LP changes as a function of the right-hand side. In this case, the parameterized version that we are interested in is defined by $\mathscr{X} \equiv \{\mathcal{P}(d) \mid d \in \mathbb{R}^m\}$ with $\mathcal{P}(d) = \{x \in \mathbb{R}^n_+ \mid Ax = d\}$, which gives the value function

$$z_{LP}(d) = \min_{x \in \mathcal{P}(d)} \quad cx \tag{1.8}$$

for all $d \in \mathbb{R}^m$. For a particular right-hand-side b, we construct a dual problem as follows

$$\max\{F(b) \mid F(d) \leq z_{LP}(d), d \in \mathbb{R}^m, F : \mathbb{R}^m \rightarrow \mathbb{R}\}. \tag{1.9}$$

As we have noted, if the primal problem has a finite optimal value, then (1.9) always has a solution F^* that is a strong dual function defined by $F^*(d) = z_{LP}(d)$ when $\mathcal{P}(d) \neq \emptyset$, and $F^*(d) = 0$ elsewhere.

We consider whether it is possible to restrict the space of functions so as to obtain a tractable method for constructing a strong dual function. There are several natural candidates for this restriction but it turns out that the simplest of those, linear functions, are adequate for this purpose. Note that

$$\begin{aligned} F(d) \leq z_{LP}(d),\ d \in \mathbb{R}^m \quad &\iff \quad F(d) \leq cx\ ,\ x \in \mathcal{P}(d),\ d \in \mathbb{R}^m \\ &\iff \quad F(Ax) \leq cx\ ,\ x \in \mathbb{R}^n_+. \end{aligned} \tag{1.10}$$

Restricting the function search space F to linear functions, the dual problem (1.9) can be rewritten as

$$\max \{ub \mid uA \leq c, u \in \mathbb{R}^m\}\ , \tag{1.11}$$

which is itself another LP, usually called the dual LP with respect to (1.6). This results in the following theorem known as the strong duality theorem for LPs.

Theorem 1.1 *For a given $b \in \mathbb{R}^m$ with a finite $z_{LP}(b)$, there exists an optimal solution u^* to the dual problem such that $F^*(d) = u^*d\ \forall d \in \mathbb{R}^m$ is a strong dual function.*

The consequences of Theorem 1.1 are enormous. This duality relation has led to efficient procedures for computing bounds, has extended our ability to perform post facto solution analysis, is the basis for procedures such as column generation and reduced cost fixing, and has yielded optimality conditions that can be used for *warm-starting* techniques that allow us either to detect optimality or to continue solving an optimization problem from an advanced starting point after having modified the input data or the parameters of the algorithm.

Optimality Conditions. For the primal-dual pair of LPs (1.6) and (1.11), if x^* and u^* are the optimal solutions respectively, a consequence of Theorem 1.1 is that we must have $cx^* = u^*b$. In other words, the optimal objective values are equal. Using this property, one can easily get a *proof of optimality* for the primal problem by dual solution. Furthermore, optimal primal-dual solution pairs have to satisfy *complementary slackness* conditions, that is, (x^*, u^*) is an optimal solutions pair if and only if

$$\begin{aligned} u_i^*(a_i x^* - b_i) &= 0 \quad \forall i \ , \\ (c_j - u^* A_j)x_j^* &= 0 \quad \forall j \ , \end{aligned} \tag{1.12}$$

where a_i is the i^{th} row and A_j is the j^{th} column of A. These conditions force the j^{th} dual constraint to be binding if the corresponding primal variable is in use, i.e., ($x_j^* > 0$).

Sensitivity Analysis. LP duality, in turn, enables us to estimate the effects of parameter changes (primarily the right-hand side and objective vectors) on the optimal solution value, usually known as *sensitivity analysis*. First of all, notice that the dual function, by definition, satisfies $F^*(d) \leq z_{LP}(d) \ \forall d \in \mathbb{R}^m$ and therefore, is a tool for both local and global sensitivity analysis.

In the case of local sensitivity analysis, we can easily determine the sensitivity of the primal-dual pair (x^*, u^*) to the parameter changes, i.e., the conditions under which they both remain optimal, by checking the feasibility and optimality requirements. In this sense, a useful attribute attached to each primal variable is the *reduced cost*, defined by the slack value $c_j - \hat{u}A_j$ of a dual constraint for some dual feasible solution $\hat{u}$. If $\hat{u}$ is optimal, the reduced cost of each inactive variable j represents the change in c_j required in order to enable this variable to be potentially active. Alternatively, it denotes the additional profit to be obtained by making this variable active. Evaluating the columns through reduced costs is the basis of column generation algorithms in large-scale optimization, where the idea is to maintain only the active columns in the master problem so that the size remains manageable.

Economic Interpretation. The dual problem also has an intuitive economic interpretation. From a buyer's perspective, the primal problem is that of minimizing the cost of purchasing a bundle of goods (or processes) satisfying some specified requirements. From a seller's perspective, the dual problem is that of maximizing the revenue, subject to satisfaction of those requirements, while remaining in the economic interest of the buyer. In a sense, the dual problem is a re-modeling of the primal problem in terms of prices for satisfying the requirements. Thus, it helps the buyer to measure the response of his total cost to a change in the level of requirements. As long as both buyer and seller act rationally, each dual variable u_i can be seen as the *marginal cost* (or *shadow price*) of supplying the requirement i. In other words, u_i represents the change in the optimal solution value per unit change in its level.

Warm Starting. On the other hand, for parameter changes that result in violation of the optimality conditions, LP duality allows us to derive warm-starting techniques. The most common solution algorithm for LPs, the *simplex method*, uses LP duality extensively to warm start the modified problem rather than solving it from scratch. This methodology is also the basis of the *dual simplex method*, which iterates over the set of feasible dual solutions. In fact, this methodology is at the core of many modern integer programming solvers, in which a series of related LPs need to be solved during the node evaluation and strong branching stages of the branch-and-cut procedure, where u^* or a simple extension of it remains dual feasible and hence, is preferred as the advanced starting point to reach the new optimum.

The Value Function. The main reason for the tractability of the LP dual can also be seen by considering the structure of the LP value function. Note that the value function, which can be rewritten as

$$z_{LP}(d) = \max \left\{ud \mid uA \leq c, u \in \mathbb{R}^m\right\}, \tag{1.13}$$

is piecewise-linear and convex over $\Omega_{LP} = \{d \in \mathbb{R}^m \mid \mathcal{P}(d) \neq \emptyset\}$ and for a right-hand-side b, a corresponding dual optimal solution u^* yields a strong dual function by Theorem 1.1 and is a *subgradient* of z_{LP} at b, that is,

$$z_{LP}(b) + u^*(d - b) \leq z_{LP}(d) \quad \forall d \in \Omega_{LP}. \tag{1.14}$$

Then, unless b is a break point of z_{LP}, u^* remains dual optimal on this linear segment (see Figure 1.1). In other words, $z_{LP}(d) = u^*d$ for all d in a sufficiently small neighborhood of b.

The fact that we can write the value function z_{LP} as in (1.13) is because the optimal dual

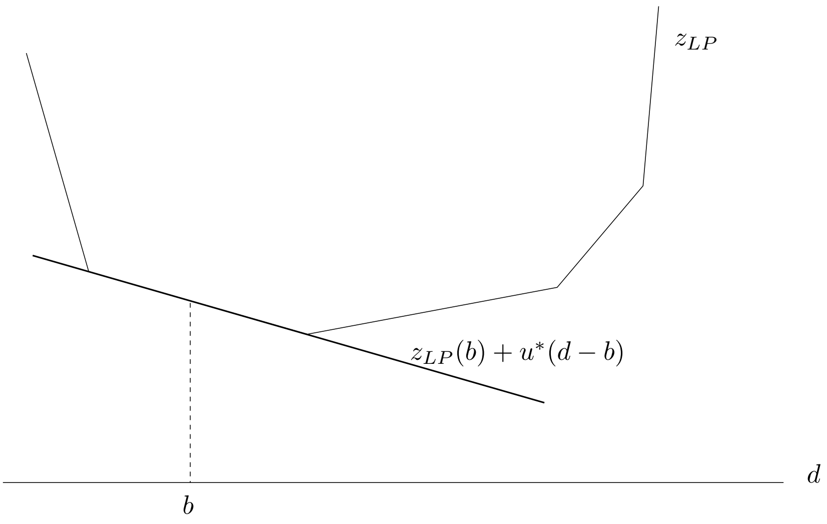


Figure 1.1: A conceptual illustration of the value function z_{LP} and its subgradient at b.

solution for a given right-hand-side yields a strong dual function. This then enables us to construct z_{LP} explicitly as a maximum of finitely many linear functions. In the case that the dual feasible region $\{\ u \in \mathbb{R}^m \mid uA \leq c\}$ is a polytope, if we let V be the set of its extreme points, then $|V|$ is finite, there exists a subgradient $v^* \in V$ for a given $b \in \mathbb{R}^m$, and we can write

$$z_{LP}(d) = \max_{v \in V}\{vd\} \quad \forall d \in \mathbb{R}^m. \tag{1.15}$$

This result is used extensively to approximate the value function in large-scale problems where knowledge of the structure of the value function of an LP is required to implemet a given method. For instance, this method of approximating the value function is the basis for *Benders decomposition* algorithms. To illustrate, assume that we are interested in an optimization problem in which the cost function includes the value function of an LP problem, such as

$$\min_{x \in U} \quad f(x) + z_{LP}(Tx)\ , \tag{1.16}$$

with $U \subseteq \mathbb{R}^n$ and $T \in \mathbb{Q}^{m \times n}$. Let F be a dual function for the value function z_{LP} and consider

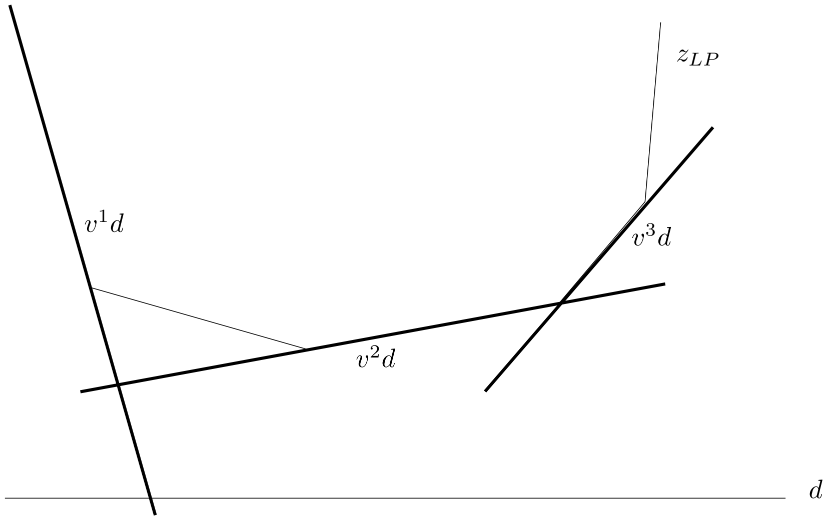


Figure 1.2: The value function z_{LP} and an approximate function $\max\{v^1d, v^2d, v^3d\}$ obtained from a set of extreme points of dual feasible region.

the following problem

$$\begin{aligned} \min \quad & f(x) + y \\ \text{s.t.} \quad & Tx = d \\ & y \geq F(d) \\ & x \in U, d \in \mathbb{R}^m \ , \end{aligned} \tag{1.17}$$

which is a relaxation of the original problem (1.16). Let (x^*, d^*) be an optimal solution of this relaxation and note that x^* is also an optimal solution of (1.16) if and only if $F(d^*) = z_{LP}(Tx^*)$. The Benders approach is to solve the original instance using (1.17) by iteratively constructing the approximation F by generating a sequence of dual functions strong for different right-hand sides through the optimal dual solutions. In a typical iteration, we have a set $K \subseteq V$, where $v \in K$ is an optimal dual solution for a right-hand side generated in the sequence. Then, since $F_v(d) = vd \ \forall d \in \mathbb{R}^m$ is a dual function for all $v \in V$, we approximate the value function by $F(d) = \max_{v \in K}\{F_v(d)\} \ \forall d \in \mathbb{R}^m$, which is also a dual function. Furthermore, this approximation can be easily represented using linear constraints, since it is piecewise-linear and convex. Consequently, the formulation (1.17) reduces to

$$\begin{aligned} \min \quad & f(x) + y \\ \text{s.t.} \quad & Tx = d \\ & y \geq F_v(d) \qquad\qquad v \in K \\ & x \in U, d \in \mathbb{R}^m \ . \end{aligned} \tag{1.18}$$

Let (x^*, d^*) be an optimal solution of (1.18) in a given iteration. Then, unless $\max_{v \in K}\{F_v(d^*)\} = z_{LP}(d^*)$, we set $K = K \cup \{v^*\}$ where v^* is the optimal dual solution that gives $z_{LP}(d^*) = v^* d^*$ and re-solve (1.18) with the updated approximation.

1.3.2 Mixed Integer Linear Programming

The success that has come from advances in LP duality can be summarized as follows: (1) the classical dual problem is just another LP and is a manageable optimization problem that enables us to derive optimality conditions and strong primal-dual relations, (2) strong dual functions, which are linear, can be easily extracted from primal solution procedures, which generate an optimal dual function as a by-product of proving optimality, (3) the value function is piecewise-linear and convex and the strong dual functions are just its subgradients, which makes it possible to do effective local sensitivity analysis, (4) the space of dual functions can be represented as a polyhedron, which makes it possible to write the value function explicitly as the maximum of finitely many linear functions. This last property means it is easy to embed approximations of the value function into larger linear optimization problems.

In contrast, none of these properties hold for mixed integer linear programs. Following the same steps as in LP case, we can parameterize the right-hand-side for the primal problem (1.7) and obtain the MILP value function $z : \mathbb{R}^m \to \mathbb{R} \cup \{\pm\infty\}$ defined as

$$z(d) = \min_{x \in \mathcal{S}(d)} cx, \tag{1.19}$$

where for a given $d \in \mathbb{R}$, $\mathcal{S}(d) = \{\mathcal{P}(d) \cap (\mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+)\}$. Similarly, a MILP dual problem is defined as

$$\max\{F(b) : F(d) \leq z(d),\ d \in \mathbb{R}^m, F : \mathbb{R}^m \to \mathbb{R}\}, \tag{1.20}$$

where b is the right-hand side for the instance of interest. Again, the infinite family of constraints ensures that we only consider dual functions that yield a valid bound for any right-hand-side.

Considering the fact that even computing the value function at a fixed right-hand side is an NP-hard optimization problem in general, computing the value function is extremely difficult. As we will discuss in detail later, our first observation is that, even though z is piecewise-linear, it is no longer convex. The subgradients do not in general yield valid dual functions and therefore, restricting F in (1.20) to linear functions does not guarantee the existence of a strong dual function. The MILP dual problem must remain as an optimization problem over a function space, which appears intractable by conventional optimization algorithms. Therefore, a primal-dual theory well integrated with practice has not yet been introduced.

It is important to note that for a dual function to be optimal to (1.20), it only has to agree with the value function at b. This means that not all optimal solutions to (1.20) provide the same quality bound for a given vector $d \neq b$. In fact, there are optimal solutions to this dual that provide arbitrarily poor estimates of the value function for right-hand-sides $d \neq b$, even when d is in a local neighborhood of b. Strong dual functions may not be easy to construct, represent or evaluate as observed in the LP case anymore. On the other hand, although there are some partial works on defining analogs to reduced costs and complementary slackness conditions, they are not generally applicable in practice and current MILP dual formulations do not allow us to introduce linear prices uniquely attached to each resource.

Due to the lack of a practical framework, discrete optimization techniques suffer from a corresponding lack of efficient techniques for integrating MILP duality into applications. One of those is exact dual methods for large-scale problems that substitute the value function of an MILP with an approximate one, as in (1.18) discussed for LP case. Such an algorithm would help to save the wasted computational effort, since it eliminates the need to evaluate the value function explicitly. On the other hand, there is little available in the literature concerning warm-starting for MILPs, which has a potential to be particularly useful. The aim of such techniques would be to keep and to use the dual information generated by the solution algorithm for the original problem in such a way that the solution time is decreased compared to solving the new problem from scratch. Warm-starting techniques can be useful for improving the efficiency of algorithms that rely on solution of a sequence of closely related MILP problems including the ones for parametric MILP problems, decomposition algorithms, analysis of MILP infeasibility, Lagrangian relaxation, stochastic programs and multi-criteria optimization problems.

1.4 Contributions

Our contribution to notion of MILP duality with this study can be summarized as follows:

- in depth review of duality and dual functions for MILPs,
- extending those results to branch-and-cut method,
- theoretical examination of the value function of an MILP with a single constraint and a procedure for evaluating this value function,
- extending some of those results to general MILPs,
- methods for approximating the value function of a general MILP,
- dual methods to utilize those approximations in large-scale/multi-stage problems, and
- developing and implementing warm-starting techniques for MILPs.

1.5 Outline of Thesis

Since dual functions are a central tool for approximating the value function, we first consider in Chapter 2 procedures for obtaining dual solutions through relaxations, transformations and primal solution algorithms and extend those to branch-and-cut method. This will allow us not only to enable the development of algorithms for large-scale instances, but also to define optimality conditions and derive sensitivity analysis tools. In Chapter 3, we look into the structure of the value function of an MILP. We analyze the value function of an MILP with a single constraint in detail and derive an algorithm to construct the complete function. Then, we present approximation methods for the value function of a general MILP to utilize dual functions, including the value functions of single constraint relaxations. We also outline dual methods that enable us to use these approximations to solve large-scale algorithms. In Chapter 4, we discuss how to make use of the information generated during primal solution procedure to solve a related problem. In order to do that, we initially state sufficient optimality conditions and local sensitivity analysis that follow directly from duality theory. Afterwards, we construct techniques for a warm-start as an advanced starting point to start the solution procedure of a modified problem. Finally we will report our implementation results for the methods derived for warm-starting.

Chapter 2

Mixed Integer Linear Programming Duality

A central notion in the study of duality is the concept of a dual function. Construction of such functions enable us to develop methods for determining the effect of modifications to the input data on the optimal solution value and for iteratively approximating the value function from below for large-scale instances. As we discussed before, it is difficult to develop a standard dual problem for MILP with properties similar to those observed in the LP case. However, we can still derive tractable methods for generating dual functions by considering relaxations of the original problem or by taking advantage of information produced as a by-product of a primal solution algorithm.

In this chapter, we review previous work on methods of constructing dual functions with an eye towards developing a framework for MILP duality that can be integrated with modern computational practice. Computational methods have evolved significantly since most of the work on integer programming duality was done and a close reexamination of this early work is needed. In addition, we extend in Section 2.2.5 the current framework of extracting dual information from primal solution algorithms to the case of branch-and-cut algorithm, which is the most commonly employed solution algorithm for MILPs today. We have attempted to make this chapter as general and self-contained as possible by extending known results from the pure integer to the mixed integer case whenever possible. We have included proofs for as many results as space would allow, concentrating specifically on results whose proofs were not easily accessible or for which we provide a generalization or alternative approach. The proofs for all results not included here can be found in the references cited.

2.1 The Subadditive Dual

As stated before, the dual (1.20) is rather general and perhaps only of theoretical interest. A natural question is whether it is possible to restrict the class of functions considered in this formulation in some reasonable way. Both linear and convex functions are natural candidates for consideration. If we take $\Upsilon^m \equiv \mathcal{L}^m$, then (1.20) reduces to $z_D = \max\{vb \mid vA \leq c, v \in \mathbb{R}^m\}$, which is the dual of the LP relaxation of the original MILP discussed earlier. Hence, this restriction results in a dual that is no longer guaranteed to produce a strong dual function (see Figure 2.2). Jeroslow [1979] showed that the optimum z_D obtained by setting $\Upsilon^m \equiv \mathcal{C}^m$ also results in the same optimal solution value obtained in the linear case.

Example 1 Consider the following MILP instance with right-hand side b:

$$\begin{aligned} z_{IP} = \min \quad & \tfrac{1}{2}x_1 + 2x_3 + x_4 \\ \text{s.t} \quad & x_1 - \tfrac{3}{2}x_2 + x_3 - x_4 = b \quad \text{and} \\ & x_1, x_2 \in \mathbb{Z}_+, x_3, x_4 \in \mathbb{R}_+. \end{aligned} \tag{2.1}$$

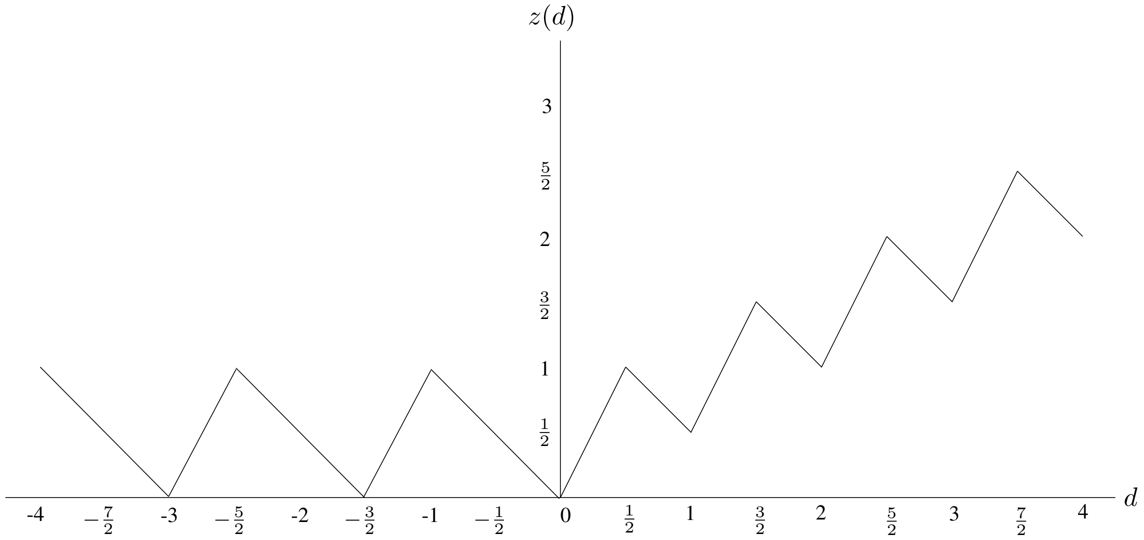


Figure 2.1: Value function of MILP from Example 1.

In this case, the value function (pictured in Figure 2.1) can be represented explicitly in the form:

$$z(d) = \begin{cases} \vdots & \vdots \\ -d - \frac{3}{2}, & -\frac{5}{2} < d \leq -\frac{3}{2} \\ 2d + 3, & -\frac{3}{2} < d \leq -1 \\ -d, & -1 < d \leq 0 \\ 2d, & 0 < d \leq \frac{1}{2} \\ -d + \frac{3}{2}, & \frac{1}{2} < d \leq 1 \\ 2d - \frac{3}{2}, & 1 < d \leq \frac{3}{2} \\ \vdots & \vdots \end{cases} . \tag{2.2}$$

We consider properties of the value function and its structure in more detail in Chapter 3. However, by checking what optimal solutions to this simple MILP instance look like as the right-hand side is varied, we can get an intuitive feel for why the value function has the shape that it does in this example. Note that the slope near zero is exactly the objective function coefficients of the continuous variables, since these are the only variables that can have positive value for values of d near zero. Furthermore, the gradients of the function alternate between these two slope values

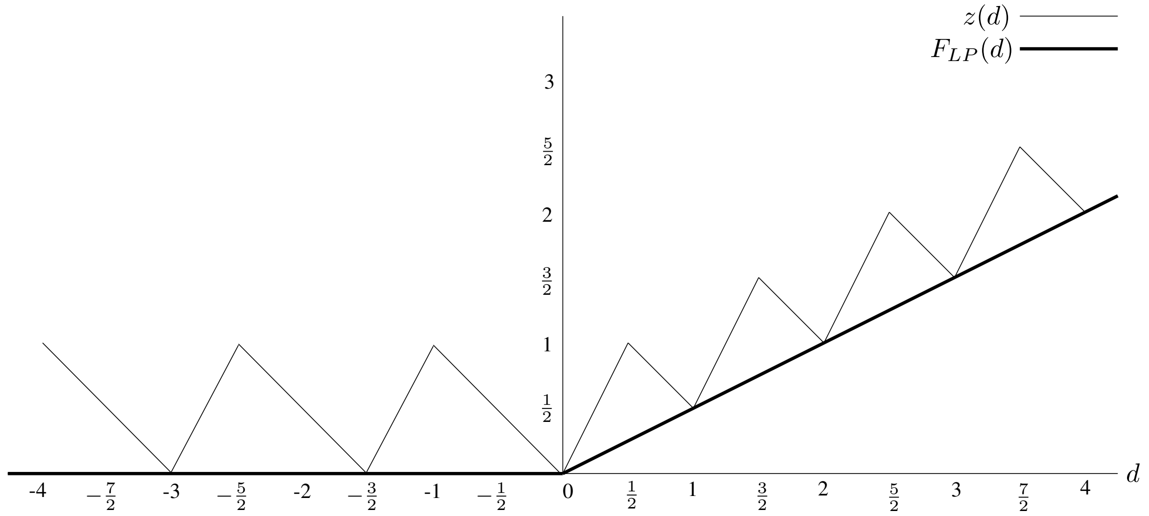


Figure 2.2: Value function of LP relaxation of problem (2.1).

moving away from zero in both directions, as the continuous variables alternate in the role of ensuring that the fractional part of the left-hand side is consistent with that of d. The coefficients of the integer variables, on the other hand, determine the breakpoints between the linear pieces.

Now consider the value function of the LP relaxation of the problem (2.1),

$$\begin{aligned} z_{LP}(d) = \max \quad & vd, \\ \text{s.t} \quad & 0 \le v \le \tfrac{1}{2}, \text{ and} \\ & v \in \mathbb{R}, \end{aligned} \tag{2.3}$$

which can be written explicitly as

$$z_{LP}(d) = \begin{cases} 0, & d \le 0 \\ \frac{1}{2}d, & d > 0 \end{cases} . \tag{2.4}$$

This dual function is shown in Figure 2.2, along with the value function of the original MILP. In this example, z_{LP} can be seen to be the best piecewise-linear, convex function bounding z from below and hence, is the optimal solution to the dual problem (1.20) when the feasible function search space is restricted to convex functions. □

In a series of papers, Johnson [1973, 1974, 1979] and later Jeroslow [1978] developed the idea of restricting Υ^m to a certain subset of $\mathcal{F}^m$ (subadditive functions). The subadditive functions are a superset of the linear functions that retain the intuitively pleasing property of "no increasing returns to scale" associated with linear functions. A strong motivation for considering this class of functions is that the value function itself is subadditive over the domain $\Omega = \{d \in \mathbb{R}^m \mid S(d) \neq \emptyset\}$ and can always be extended to a subadditive function on all of $\mathbb{R}^m$ (see Theorem 2.5). This means that this restriction does not reduce the strength of the dual (1.20). To see why the value function is subadditive, let $d_1, d_2 \in \Omega$ and suppose $z(d_i) = cx_i$ for some $x_i \in \mathcal{S}(d_i), i = 1, 2$. Then, $x_1 + x_2 \in \mathcal{S}(d_1 + d_2)$ and hence $z(d_1) + z(d_2) = c(x_1 + x_2) \geq z(d_1 + d_2)$.

Recall from the set of relations (1.10) that we can rewrite the MILP dual problem (1.20) as

$$z_D = \max \{F(b) : F(Ax) \leq cx, \; x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+, \; F \in \Upsilon^m\}. \tag{2.5}$$

Then, if $\Upsilon^m \equiv \Gamma^m \equiv \{F \in \mathcal{F}^m \mid F(0) = 0\}$, we can rewrite (2.5) as the *subadditive dual*

$$\begin{aligned} z_D = \max \quad & F(b) \\ & F(a^j) \leq c_j \quad \forall j \in I, \\ & \bar{F}(a^j) \leq c_j \quad \forall j \in C, \text{ and} \\ & F \in \Gamma^m, \end{aligned} \tag{2.6}$$

where a^j is the j^{th} column of A and the function $\bar{F}$ is defined by

$$\bar{F}(d) = \limsup_{\delta \to 0^+} \frac{F(\delta d)}{\delta} \quad \forall d \in \mathbb{R}^m. \tag{2.7}$$

Here, $\bar{F}$, first used by Gomory and Johnson [1972] in the context of cutting plane algorithms, is the *upper d-directional derivative* of F at zero. The next result reveals the relation between F and $\bar{F}$.

Theorem 2.1 *(Johnson [1974], Jeroslow [1978], Nemhauser and Wolsey [1988]) If* $F \in \Gamma^m$*,* *then* $\forall d \in \mathbb{R}^m$ *with* $\bar{F}(d) < \infty$ *and* $\lambda \geq 0$, $F(\lambda d) \leq \lambda \bar{F}(d)$.

Proof. Let $\lambda > 0$ and $\mu > 0$. Setting $q = \mu - \lfloor \mu \rfloor$, we have

$$\begin{aligned} F(\lambda d) = F\left(\frac{\mu \lambda d}{\mu}\right) = F\left(\frac{\lfloor \mu \rfloor \lambda d}{\mu} + \frac{q\lambda d}{\mu}\right) &\leq \lfloor \mu \rfloor F\left(\frac{\lambda d}{\mu}\right) + F\left(\frac{q\lambda d}{\mu}\right) \\ &= \mu F\left(\frac{\lambda d}{\mu}\right) + F\left(\frac{q\lambda d}{\mu}\right) - qF\left(\frac{\lambda d}{\mu}\right), \end{aligned}$$

where the inequality follows from the fact that $F \in \Gamma^m$. Now, letting $\delta = \frac{1}{\mu}$, we get

$$F(\lambda d) \leq \frac{F(\delta \lambda d)}{\delta} + q\delta\left(\frac{F(q\delta\lambda d)}{q\delta} - \frac{F(\delta\lambda d)}{\delta}\right). \tag{2.8}$$

By taking the limit as $\delta \to 0^+$, we obtain

$$F(\lambda d) \leq \bar{F}(\lambda d). \tag{2.9}$$

Finally, note that

$$\bar{F}(\lambda d) = \limsup_{\delta \to 0^+} \frac{F(\delta(\lambda d))}{\delta} = \limsup_{\delta\lambda \to 0^+} \frac{\lambda F(\delta \lambda d)}{\delta \lambda} = \lambda \bar{F}(d). \tag{2.10}$$

□

Example 2 Consider the d-directional derivative of the value function for the MILP (2.1), shown in Figure 2.3:

$$\bar{z}(d) = \begin{cases} -d, & d \leq 0 \\ 2d, & d > 0 \end{cases}. \tag{2.11}$$

Note that $\bar{z}$ is a piecewise linear convex function whose directional derivatives near the origin coincide with that of z. As we pointed out in Example 1, these directional derivatives are determined by the coefficients of the continuous variables in (2.1). □

The use of $\bar{F}$ is required in (2.6) due to the presence of the continuous variables and is not needed for pure integer programs. Intuitively, the role of $\bar{F}$ is to ensure that solutions to (2.6) have gradients that do not exceed those of the value function near zero, since the subadditivity of F

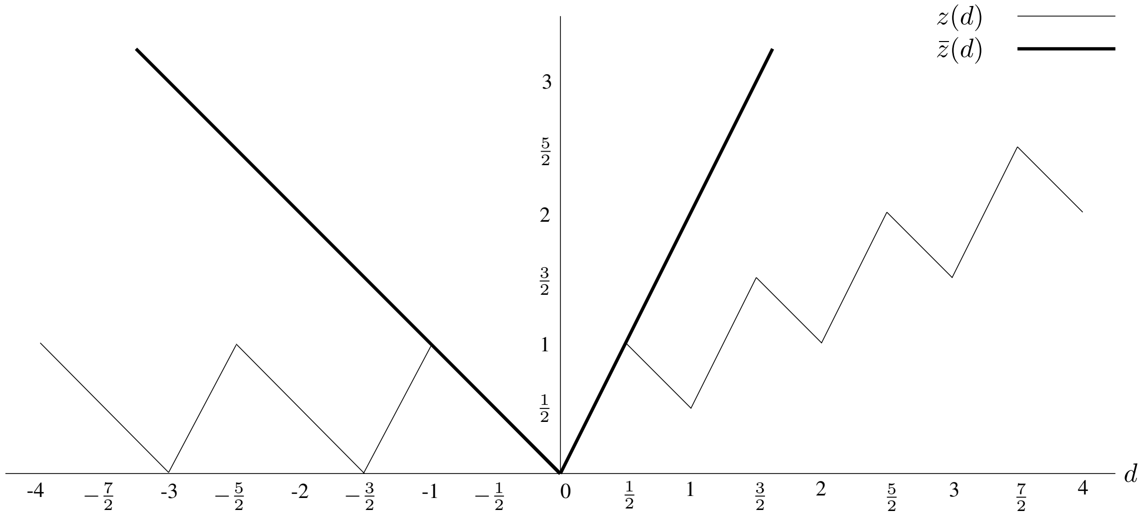


Figure 2.3: Directional derivative of the value function of problem (2.1).

alone is not enough to ensure this in the case of MILP. We now show formally that (2.6) is in fact a valid dual problem.

Theorem 2.2 *(Weak Duality by Jeroslow [1978, 1979]) Let x be a feasible solution to the MILP (1.7) and let F be a feasible solution to the subadditive dual (2.6). Then, $F(b) \leq cx$.*

Proof. Let x be a given feasible solution to the MILP (1.7). Note that

$$\begin{aligned} F(b) &= F(Ax) \\ &\leq F(\sum_{j\in I} a^j x_j) + F(\sum_{j\in C} a^j x_j) \\ &\leq \sum_{j\in I} F(a^j) x_j + \sum_{j\in C} \bar{F}(a^j) x_j \\ &\leq cx. \end{aligned}$$

The first inequality follows from the subadditivity of F. Next, $F(\sum_{j\in I} a^j x_j) \leq \sum_{j\in I} F(a^j)x_j$, since F is subadditive, $F(0) = 0$ and $x_j \in \mathbb{Z}_+, j \in I$. Similarly, $F(\sum_{j\in C} a^j x_j) \leq \sum_{j\in C} F(a^j x_j)$ $\leq \sum_{j\in C} \bar{F}(a^j)x_j$, since $\bar{F}(0) = 0$ and $F(a^j x_j) \leq \bar{F}(a^j)x_j,\ x_j \in \mathbb{R}_+,\ j \in C$ by Theorem 2.1. Therefore, the second inequality holds. For the last inequality, $F(a^j) \leq c_j,\ j \in I$ and $\bar{F}(a^j) \leq c_j,\ j \in C$ by dual feasibility and x_j is nonnegative for all $j \in N$ by primal feasibility.□

Example 3 For the MILP (2.1), the subadditive dual problem is

$$\begin{array}{rrl}\max & F(b) & \\ & F(1) & \leq \frac{1}{2} \\ & F(-\frac{3}{2}) & \leq 0 \\ & \bar{F}(1) & \leq 2 \\ & \bar{F}(-1) & \leq 1 \\ & F & \in \Gamma^1. \end{array} \quad . \tag{2.12}$$

As described above, the last two constraints require that the slope of F going away from the origin (the d-directional derivative) be less than or equal to that of the value function, whereas the first two constraints require that $F(1)$ and $F(-\frac{3}{2})$ not exceed $z(1)$ and $z(-\frac{3}{2})$, respectively. Note that in this example, the constraint $\bar{F}(-1) \leq 1$ is actually equivalent to the constraint $F(-1) \leq 1$, but replacing $\bar{F}(1) \leq 2$ with $F(1) \leq 2$ results in the admission of invalid dual functions.

If we require integrality of all variables in (2.1), then the value function becomes that shown in Figure 2.4, defined only at discrete values of the right-hand side d. In this case, $\bar{F}$ is replaced by F in (2.12) and the third constraint becomes redundant. This can be seen by the fact that x_3 cannot take on a positive value in any optimal solution to the pure integer restriction of (2.1). □

Although the value function itself yields an optimal solution to the subadditive dual of any given MILP, irrespective of the value of the original right-hand side b, the set of all dual functions that are optimal to (2.6) can be affected dramatically by the initial value of b considered. This is because F is required to agree with the value function only at b and nowhere else. In the following example, we consider optimal solutions to (2.12) for different values of b.

Example 4 Consider optimal solutions to (2.6) for the MILP (2.1) for different values of b.

i. $F_1(d) = \frac{d}{2}$ is an optimal dual function for $b \in \{0, 1, 2, \dots\}$ (see Figure 2.2),

ii. $F_2(d) = 0$ is an optimal dual function for $b \in \{\dots, -3, -\frac{3}{2}, 0\}$ (see Figure 2.2).

iii. $F_3(d) = \max\{\frac{1}{2}\lceil d - \frac{\lceil\lceil d\rceil - d\rceil}{4}\rceil, 2d - \frac{3}{2}\lceil d - \frac{\lceil\lceil d\rceil - d\rceil}{4}\rceil\}$ is an optimal dual function for $b \in$

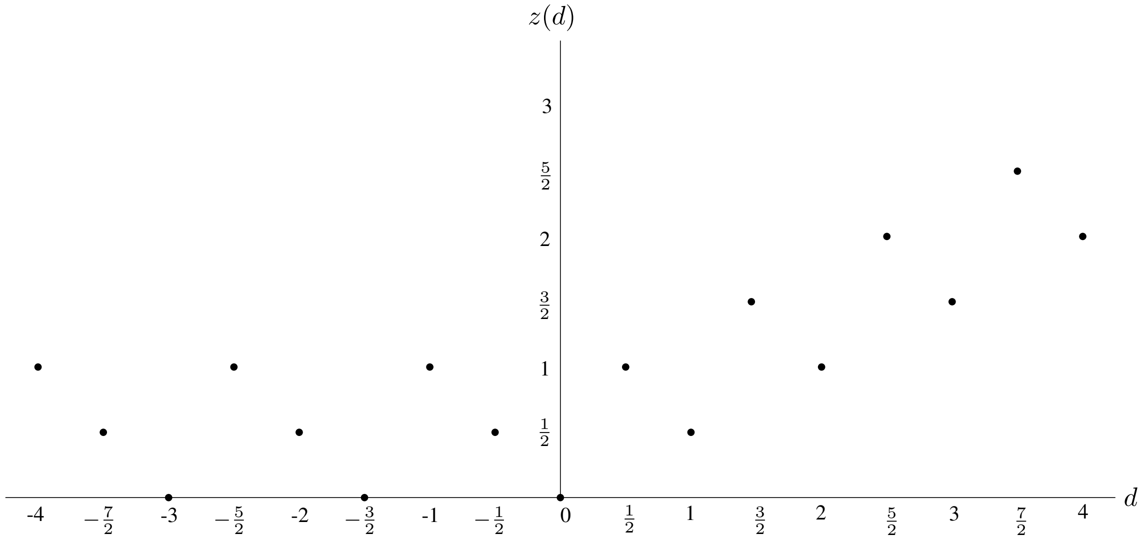


Figure 2.4: Value function of problem (2.1) with all variables integer.

$\{[0, \frac{1}{4}] \cup [1, \frac{5}{4}] \cup [2, \frac{9}{4}] \cup \ldots\}$ (see Figure 2.5).

iv. $F_4(d) = \max \frac{3}{2}\lceil \frac{2d}{3} - \frac{2\lceil \lceil \frac{2d}{3} \rceil - \frac{2d}{3} \rceil}{3} \rceil - d, -\frac{3}{4}\lceil \frac{2d}{3} - \frac{2\lceil \lceil \frac{2d}{3} \rceil - \frac{2d}{3} \rceil}{3} \rceil + \frac{d}{2}\}$ is an optimal dual function for $b \in \{\cdots \cup [-\frac{7}{2}, -3] \cup [-2, -\frac{3}{2}] \cup [-\frac{1}{2}, 0]\}$ (see Figure 2.5). □

As in LP duality, weak duality yields results concerning the relationship between primal and dual when no finite optimum exists. Before proving the main corollary, we need the following important lemma.

Lemma 2.3 *For the MILP (1.7) and its subadditive dual (2.6), the following statements hold:*

i. The primal problem is unbounded if and only if $b \in \Omega$ and $z(0) < 0$.

ii. The dual problem is infeasible if and only if $z(0) < 0$.

Proof. First, note that $0 \in \Omega$ and $z(0) \leq 0$, since $x = 0$ is a feasible solution to the MILP (1.7) with right-hand side 0.

i. If $b \in \Omega$ and $z(0) < 0$, then there exist $\bar{x} \in S$ and $\hat{x} \in S(0)$ with $c\hat{x} < 0$. Then $\bar{x} + \lambda\hat{x} \in S$ for all $\lambda \in \mathbb{R}_+$ and it follows that λ can be chosen to make $z(b)$ arbitrarily small. Conversely, if $b \in \Omega$ and $z(0) = 0$, then we must also have that $\min\{cx \mid Ax = 0, x \in \mathbb{R}^n_+\} = 0$.

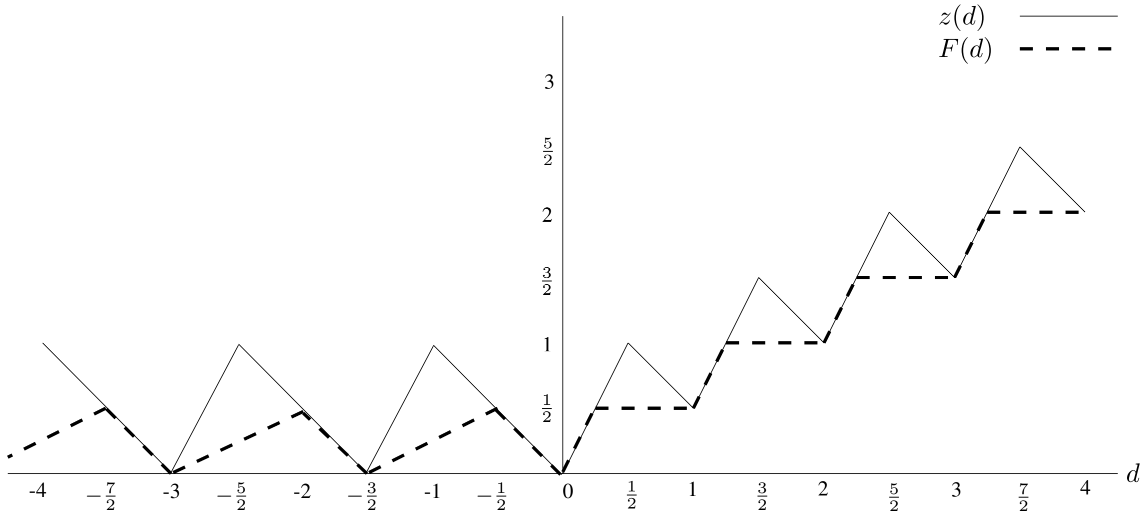


Figure 2.5: Observe that $F(d) = \max\{F_3(d), F_4(d)\}$ is an optimal dual function for (2.12) for some values of b and only feasible otherwise.

Otherwise, there must exist an $\hat{x} \in \mathbb{Q}^n_+$ for which $A\hat{x} = 0$ and $c\hat{x} < 0$, which can be scaled to yield an integer solution to (1.7) with right-hand side 0, contradicting the assumption that $z(0) = 0$. Since no such vector exists, the LP relaxation of (1.7), and hence the MILP itself, must be bounded.

ii. If $z(0) = 0$, then $\min\{cx \mid Ax = 0, x \in \mathbb{R}^n_+\} = \max\{v0 : vA \leq c, v \in \mathbb{R}^m\} = 0$ (see the proof of part 1 above) and therefore, (2.6) is feasible by setting $F(d) = v^*d \; \forall \; d \in \mathbb{R}^m$, where v^* is the optimal dual solution. This implies that if the dual is infeasible, then $z(0) < 0$. If $z(0) < 0$, on the other hand, the dual cannot be feasible since any feasible solution F has to satisfy $F(0) = 0$ and this would contradict weak duality. □

Corollary 2.4 *For the MILP (1.7) and its subadditive dual (2.6),*

i. If the primal problem (resp., the dual) is unbounded then the dual problem (resp., the primal) is infeasible.

ii. If the primal problem (resp., the dual) is infeasible, then the dual problem (resp., the primal) is infeasible or unbounded.

Proof. i. This part follows directly from Theorem 2.2.

ii. Assume that the primal problem is infeasible. Then there are two cases. If $z(0) < 0$, the dual is infeasible by Lemma 2.3. Otherwise, by LP duality, $\exists\ v \in \mathbb{R}^m$ with $vA \leq c$. Let $F_1 \in \Gamma^m$ be defined by $F_1(d) = vd,\ \forall\ d \in \mathbb{R}^m$. Note that $\bar{F}_1 = F_1$. Next, consider the value function $F_2(d) = \min\{x_0 : Ax + x_0 d = d,\ x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+,\ x_0 \in \mathbb{Z}_+\}$. F_2 is defined and finite for all $d \in \mathbb{R}^m$ since $x = 0$ with $x_0 = 1$ is a feasible solution for any right-hand side. Therefore, $F_2 \in \Gamma^m$. Furthermore, for any $j \in I$, $F_2(a^j) \leq 0$, since e^j (the j^{th} unit vector) together with $x_0 = 0$ is a feasible solution to the corresponding problem. On the other hand, for any $j \in C$ and $\delta > 0$, $F_2(\delta a^j) \leq 0$ due to the fact that $x = \delta e^j$ and $x_0 = 0$ is feasible. Thus, $\bar{F}_2(a^j) \leq 0,\ \forall\ j \in C$. In addition, $F_2(b) = 1$ since there cannot be an optimal solution with $x_0 = 0$ as a consequence of $\mathcal{S} = \emptyset$. Then, observe that for any scalar $\lambda \in \mathbb{R}_+$, $F_1 + \lambda F_2$ is dual feasible to (2.6), which means that the dual is unbounded as λ can be chosen arbitrarily large.

If the dual problem is infeasible, then, by Lemma 2.3, $z(0) < 0$ and the primal problem□is unbounded if $b \in \Omega$ and infeasible otherwise.

Before moving on to prove strong duality, we need the following theorem that states that any given MILP can be "extended" to one that is feasible for all right-hand sides and whose value function agrees with that of the original MILP for all right-hand sides $d \in \Omega$.

Theorem 2.5 *(Blair and Jeroslow [1977]) For the MILP (1.7), there exists an extension* $z_e(d) = \min\{c_e x : A_e x = d, x \in \mathbb{Z}^l_+ \times \mathbb{R}^{k-l}_+\}$*, with* c_e *and* A_e *obtained by adding new coefficients and columns to c and A, such that* $z_e(d) = z(d)\ \forall\ d \in \Omega$ *and* $z_e(d) < \infty\ \forall\ d \in \mathbb{R}^m$.

We will not give the proof here, but note that the idea depends on iteratively adding columns orthogonal to the span of the columns of A with objective function coefficients chosen so that $z_e(d) = z(d)$ whenever $z(d) < \infty$. The following result then shows formally that the dual (2.6) is strong.

Theorem 2.6 *(Strong duality by Jeroslow [1978, 1979], Wolsey [1981]) If the primal problem (1.7) (resp., the dual) has a finite optimum, then so does the dual problem (2.6) (resp., the primal) and they are equal.*

Proof. Note that if the primal or the dual has a finite optimum, then Corollary 2.4 requires the other also to have a finite optimum. Now, we prove the claim by verifying that the value function z (or an extension to z) is a feasible dual function whenever the primal has a finite optimum.

i. $\Omega \equiv \mathbb{R}^m$: In this case, $z \in \Gamma^m$, and with a similar argument in the second part of the proof of Corollary 2.4, z is feasible to the dual problem.

ii. $\Omega \subset \mathbb{R}^m$: By Theorem 2.5, $\exists\, z_e \in \Gamma^m$ with $z_e(d) = z(d)\ \forall\, d \in \Omega$ and $z_e(d) < \infty\ \forall d \in \mathbb{R}^m$. By construction, z_e must satisfy the constraints of the dual of the original MILP (1.7), since the dual of the extended MILP from Theorem 2.5 includes the constraints of (2.6) ($I_e \supseteq I$ and $N_e \supseteq N$). Therefore, z_e is feasible to the dual of the original MILP and hence, this dual has an optimal solution value of $z_e(b) = z(b)$. □

One can further use the strong duality property of (2.6) to derive a generalization of Farkas' Lemma. This result is stated more formally in the following corollary.

Corollary 2.7 *(Blair and Jeroslow [1982]) For the MILP (1.7), exactly one of the following holds:*

i. $\mathcal{S} \neq \emptyset$

ii. There is an $F \in \Gamma^m$ *with* $F(a^j) \leq 0\, \forall j \in I$, $\bar{F}(a_j) \leq 0\, \forall j \in C$ *and* $F(b) > 0$.

Proof. The proof follows directly from applying Corollary 2.4 and Theorem 2.6 to the MILP (1.7) with $c = 0$. □

The subadditive dual (2.6) can also be used to extend familiar concepts such as *reduced costs* and the *complementary slackness* conditions to MILPs. For a given optimal solution F^* to (2.6), the reduced costs can be defined as $c_j - F^*(a^j)$ for $j \in I$ and $c_j - \bar{F}^*(a^j)$ for $j \in C$. These reduced costs have an interpretation similar to that in the LP case, except that we do not have the same concept of "sensitivity ranges" within which the computed bounds are exact. Complementary slackness conditions can be stated as follows.

Theorem 2.8 *(Jeroslow [1978], Johnson [1979], Bachem and Schrader [1980], Wolsey [1981]) For a given right-hand side* b*, let* x^* *and* F^* *be feasible solutions to to the primal problem (1.7)* *and the subadditive dual problem (2.6). Then,* x^* *and* F^* *are optimal if and only if*

$$\begin{aligned} x_j^*(c_j - F^*(a^j)) &= 0, \ \forall j \in I, \\ x_j^*(c_j - \bar{F}^*(a^j)) &= 0, \ \forall j \in C, \ \textit{and} \\ F^*(b) &= \sum_{j \in I} F^*(a^j)x_j^* + \sum_{j \in C} \bar{F}^*(a^j)x_j^* \end{aligned} \tag{2.13}$$

Proof. If x^* and F^* are optimal, then, from the properties of F^* and strong duality,

$$F^*(b) = F^*(Ax^*) = \sum_{j \in I} F^*(a^j)x_j^* + \sum_{j \in C} \bar{F}^*(a^j)x_j^* = cx^*. \tag{2.14}$$

Then, we have

$$\sum_{j \in I} x_j^*(c_j - F^*(a^j)) + \sum_{j \in C} x_j^*(c_j - \bar{F}^*(a^j)) = 0.$$

By primal and dual feasibility, $x_j^* \geq 0 \ \forall j \in N,\ c_j - F^*(a^j) \geq 0 \ \forall j \in I$ and $c_j - \bar{F}^*(a^j) \geq 0 \ \forall j \in C$. Therefore, (2.13) holds.

On the other hand, if the conditions (2.13) are satisfied, then (2.14) holds, which in turn, yields $F^*(b) = cx^*$. □

These conditions, if satisfied, yield a certificate of optimality for a given primal-dual pair of feasible solutions. They can further be used to develop an integer programming analog of the well-known primal-dual algorithm for linear programming. Llewellyn and Ryan [1993] give the details of one such algorithm.

The subadditive duality framework also allows the use of subadditive functions to obtain inequalities valid for the convex hull of $\mathcal{S}$. In fact, subadditive functions alone can, in theory, yield a complete description of $\text{conv}(\mathcal{S})$. It is easy to see that for any $d \in \Omega$ and $F \in \Gamma^m$ with $\bar{F}(a^j) < \infty \ \forall j \in C$, the inequality

$$\sum_{j \in I} F(a^j)x_j + \sum_{j \in C} \bar{F}(a^j)x_j \ \geq \ F(d) \tag{2.15}$$

is satisfied for all $x \in \mathcal{S}(d)$. The proof follows the same steps as that of weak duality, with x restricted to be in $\mathcal{S}(d)$. Furthermore, the following result shows that any valid inequality is either equivalent to or dominated by an inequality in the form of (2.15).

Theorem 2.9 *(Johnson [1973], Jeroslow [1978]) For the MILP (1.7) and $\pi \in \mathbb{R}^n$, $\pi_0 \in \mathbb{R}$, we have that $\pi x \geq \pi_0 \ \forall x \in \mathcal{S}$ if and only if there is an $F_\pi \in \Gamma^m$ such that*

$$\begin{aligned} F_\pi(a^j) &\leq \pi_j && \forall j \in I, \\ \bar{F}_\pi(a^j) &\leq \pi_j && \forall j \in C \ \textit{and}, \\ F_\pi(b) &\geq \pi_0. \end{aligned} \tag{2.16}$$

Proof. First assume that $\pi \in \mathbb{R}^n$ and $\pi_0 \in \mathbb{R}$ are given such that $\pi x \geq \pi_0 \ \forall x \in \mathcal{S}$. Consider the MILP

$$z_\pi = \min \ \{\pi x \mid x \in \mathcal{S}\}. \tag{2.17}$$

Clearly, $z_\pi \geq \pi_0$ because otherwise, there exists an $\bar{x} \in \mathcal{S}$ with $\pi \bar{x} < \pi_0$. Applying Theorem 2.6 to (2.17), we find that there must be a dual feasible function F_π satisfying (2.16).

Conversely, assume that there exists an $F_\pi \in \Gamma^m$ satisfying (2.16) for a given $\pi \in \mathbb{R}^n$ and $\pi_0 \in \mathbb{R}$. Then F_π is also feasible to the subadditive dual of (2.17) and from weak duality, $\pi x \geq F_\pi(b) \geq \pi_0$ for all $x \in \mathcal{S}$. □

Example 5 The subadditive dual function $F_3(d)$ in Example 4 is feasible to (2.12). Since $F_3(1) = \frac{1}{2}$, $F_3(-\frac{3}{2}) = -\frac{1}{2}$, $\bar{F}_3(1) = 2$, $\bar{F}_3(-1) = 1$, then

$$\frac{x_1}{2} - \frac{x_2}{2} + 2x_3 + x_4 \geq F_3(b) \tag{2.18}$$

is a valid inequality for (2.1). □

As an extension to this theorem, Bachem and Schrader [1980] showed that the convex hull of $\mathcal{S}$ can be represented using only subadditive functions and that rationality of A is enough to ensure the existence of such a representation, even if the convex hull is unbounded.

Theorem 2.10 *(Jeroslow [1978], Blair [1978], Bachem and Schrader [1980]) For any $d \in \Omega$,*

$$conv(\mathcal{S}(d)) = \{x : \sum_{j \in I} F(a^j)x_j + \sum_{j \in C} \bar{F}(a^j)x_j \ \geq \ F(d), \ F \in \Gamma^m, \ x \geq 0\}. \tag{2.19}$$

For a fixed right-hand side, it is clear that only finitely many subadditive functions are needed to obtain a complete description, since every rational polyhedron has finitely many facets. In fact, Wolsey [1979] showed that for PILPs, there exists a finite representation that is valid for all right-hand sides.

Theorem 2.11 *(Wolsey [1979]) For a PILP in the form (1.7), there exist finitely many subadditive functions $F_i, \ i = 1, \ldots, k$, such that*

$$conv(\mathcal{S}(d)) = \{x : \sum_{j=1}^{n} F_i(a^j)x_j \geq F_i(d), \ i = 1, \ldots, k, x \geq 0\} \tag{2.20}$$

for any $d \in \Omega$.

Proof. Wolsey [1979] showed that when $A \in \mathbb{Z}^{m \times n}$, there exists finitely many subadditive functions $F_i, \ i = 1, \ldots, k$, such that

$$conv(\mathcal{S}(d)) = \{x : Ax = d, \ \sum_{j=1}^{n} F_i(a^j)x_j \geq F_i(d), \ i = 1, \ldots, k, x \geq 0\} \ \forall \, d \in \mathbb{Z}^m. \tag{2.21}$$

However, the assumption that $A \in \mathbb{Z}^{m \times n}$ is without loss of generality, since A can be scaled appropriately. After scaling, we must have $\Omega \subseteq \mathbb{Z}^m$ and the result follows. □

Finally, it is possible to show not only that any facet can be generated by a subadditive function, but that this is true of any *minimal inequality*. Recall that $\pi \in \mathbb{R}^m$ and $\pi_0 \in \mathbb{R}$ define a minimal valid inequality if there is no other valid inequality defined by $\hat{\pi} \in \mathbb{R}^m$ and $\hat{\pi}_0 \in \mathbb{R}$ such that $\hat{\pi}_j \leq \pi_j$ for all $j \in N$ and $\hat{\pi}_0 \geq \pi_0$. Although the next theorem was originally stated for either rational constraint matrices (Johnson [1974], Blair [1978]) or bounded feasible regions (Jeroslow [1979]), Bachem and Schrader [1980] showed that the same result holds without any restriction on $\mathcal{S}$.

Theorem 2.12 *(Bachem and Schrader [1980]) If $\pi \in \mathbb{R}^m$ and $\pi_0 \in \mathbb{R}$ define a minimal valid inequality for the MILP (1.7), then there is an $F \in \Gamma^m$ such that*

$$\begin{aligned} F(a^j) &= \pi_j = F(b) - F(b - a^j) && \forall j \in I, \\ \bar{F}(a^j) &= \pi_j = \lim_{\delta \to 0^+} \tfrac{F(b) - F(b - \delta a^j)}{\delta} && \forall j \in C \ \ \textit{and}, \\ F(b) &= \pi_0. \end{aligned} \tag{2.22}$$

The converse of Theorem 2.12 holds for any subadditive function that is the value function for the MILP (1.7) with objective function π, where $\pi_0 = \min\{\pi x \mid x \in \mathcal{S}\}$ (as in (2.17)).

2.2 Constructing Dual Functions

It is reasonable to conclude that neither the general dual problem (1.20) nor the subadditive dual problem (2.6) can be formulated as manageable mathematical program solvable directly using current technology. However, there are a number of methods we can use to obtain feasible (and in some cases optimal) dual functions indirectly. We focus here on dual functions that provide bounds for instances of a given MILP after modifying the right-hand side, since these are the ones about which we know the most. Such dual functions are intuitive because they allow us to extend traditional notions of duality from the realm of linear programming to that of integer programming. However, we emphasize that they are not the only dual functions of potential interest in practice. Dual functions that accommodate changes to the objective function are also of interest in many applications, particularly decomposition algorithms. Similarly, such functions that allow changes to the constraint matrix are closely related to those for studying the right-hand side, but may also yield further insight.

Dual functions of the right-hand side can be grouped into three categories: (1) those obtained from known families of relaxations, (2) those obtained as a by-product of a primal solution algorithm, such as branch-and-cut, and (3) those constructed explicitly in closed form using a finite procedure. In Sections 2.2.1 through 2.2.3, we discuss methods for obtaining dual functions from relaxations. In Section 2.2.4, we discuss a method by which the subadditive dual of a bounded

PILP can be formulated as a linear program. Finally, in Section 2.2.5, we discuss how to obtain a dual function as a by-product of the branch-and-cut algorithm, the method used most commonly in practice for solving MILPs. We will discuss the methods of explicitly constructing the value function of an MILP later in Chapter 3.

2.2.1 Cutting Plane Method

Cutting plane algorithms are a broad class of methods for obtaining lower bounds on the optimal solution value of a given MILP by iteratively generating inequalities valid for the convex hull of $\mathcal{S}$ (called *cutting planes* or *cuts*). The procedure works by constructing progressively tighter polyhedral approximations of $conv(\mathcal{S})$, over which a linear program is then solved to obtain a bound. To be more precise, in iteration k, the algorithm solves the following linear program:

$$\begin{aligned} \min \quad & cx \\ \text{s.t.} \quad & Ax = b \\ & \Pi x \geq \Pi_0 \\ & x \geq 0, \end{aligned} \tag{2.23}$$

where $\Pi \in \mathbb{R}^{k \times n}$ and $\Pi_0 \in \mathbb{R}^k$ represents the cutting planes generated so far. At the time of generation, each of the valid inequalities is constructed so as to eliminate a portion of the feasible region of the current relaxation that contains the current solution to (2.23), but does not contain any feasible solutions to the original MILP.

As noted earlier, the LP dual of (2.23), i.e.,

$$\begin{aligned} \max \qquad & vb + w\Pi_0 \\ & vA + w\Pi \leq c \\ & v \in \mathbb{R}^m, w \in \mathbb{R}^k_+, \end{aligned} \tag{2.24}$$

is also a dual problem for the original MILP, but does not yield a dual function directly because the cutting planes generated may only be valid for the convex hull of solutions to the original

MILP and not for instances with a modified right-hand side. However, one can extract such a dual function if it is possible to systematically modify each cut to ensure validity after replacing the original right-hand side b with a new right-hand side d. Assuming that a subadditive representation (2.15) of each cut is known, the i^{th} cut can be expressed parametrically as a function of the right-hand side $d \in \mathbb{R}^m$ in the form

$$\sum_{j \in I} F_i(\sigma_i(a^j))x_j + \sum_{j \in C} \bar{F}_i(\bar{\sigma}_i(a^j))x_j \;\geq\; F_i(\sigma_i(d)), \tag{2.25}$$

where F_i is the subadditive function representing the cut, and the functions $\sigma_i, \bar{\sigma}_i : \mathbb{R}^m \rightarrow \mathbb{R}^{m+i-1}$ are defined by

- $\sigma_1(d) = \bar{\sigma}_1(d) = d$,
- $\sigma_i(d) = [\, d \;\; F_1(\sigma_1(d)) \;\; \ldots \;\; F_{i-1}(\sigma_{i-1}(d)) \,]$ for $i \geq 2$, and
- $\bar{\sigma}_i(d) = [\, d \;\; \bar{F}_1(\bar{\sigma}_1(d)) \;\; \ldots \;\; \bar{F}_{i-1}(\bar{\sigma}_{i-1}(d)) \,]$ for $i \geq 2$.

Furthermore, if (v^k, w^k) is a feasible solution to (2.24) in the k^{th} iteration, then the function

$$F_{CP}(d) = v^k d + \sum_{i=1}^{k} w_i^k F_i(\sigma_i(d)) \tag{2.26}$$

is a feasible solution to the subadditive dual problem (2.6).

As noted earlier, Wolsey [1981] showed how to construct a dual function optimal to the subadditive dual for a given PILP using the Gomory fractional cutting plane algorithm under the assumption that cuts are generated using a method guaranteed to yield a sequence of LPs with lexicographically increasing solution vectors (this method is needed to guarantee termination of the algorithm in a finite number of steps with either an optimal solution or a proof that original problem is infeasible). In Gomory's procedure, the subadditive function F_i, generated for iteration i, has the following form

$$F_i(d) = \left\lceil \sum_{k=1}^{m} \lambda_k^{i-1} d_k + \sum_{k=1}^{i-1} \lambda_{m+k}^{i-1} F_k(d) \right\rceil \quad \text{where } \lambda^{i-1} = (\lambda_1^{i-1}, \ldots, \lambda_{m+i-1}^{i-1}) \geq 0. \tag{2.27}$$

Assuming that $b \in \Omega$, $z(b) > -\infty$, and that the algorithm terminates after k iterations, the function F_G defined by

$$F_G(d) = v^k d + \sum_{i=1}^{k} w_i^k F_i(d) \tag{2.28}$$

is optimal to the subadditive dual problem (2.6).

In practice, it is generally not computationally feasible to determine a subadditive representation for each cut added to the LP relaxation. However, since our goal is simply to ensure the validity of each cut after modification of the right-hand side, an alternative approach that is feasible for some classes of valid inequalities is simply to track the dependency of each cut on the original right-hand side in some other way. If this information can be functionally encoded, as it is with the subadditive representation, the right-hand side of each cut can be modified to make it valid for new instances and these functions can be used to obtain a dual function similar in form to (2.26). As an example of this, Schrage and Wolsey [1985] showed how to construct a function tracking dependency on the right-hand side for cover inequalities by expressing the right-hand side of a cut of this type as an explicit function of the right-hand side of the original knapsack constraint. To illustrate, suppose that $\pi \in \mathbb{R}^n$ and $\pi_0 \in \mathbb{R}$ is such that $\pi \geq 0$ and $\pi_0 \geq 0$. We define $U \subseteq \{1, \ldots, n\}$ to be a *cover* if $\sum_{j \in U} \pi_j > \pi_0$. It is then well-known that $\sum_{j \in U} x_j \leq |U| - 1$ for all $x \in \{0,1\}^n$ satisfying $\pi x \leq \pi_0$. The following proposition shows how to modify the given inequality so that it remains valid if π_0 is changed to $\bar{\pi} \in \mathbb{R}$.

Theorem 2.13 *(Schrage and Wolsey [1985]) Let $\pi_v = \max\{\pi_j \mid j \in U\}$ for a given knapsack constraint with nonnegative parameters and a cover U. Then,*

$$\sum_{j \in U} x_j \leq \left\lfloor |U| - \frac{\sum_{j \in U} \pi_j - \bar{\pi}_0}{\pi_v} \right\rfloor \tag{2.29}$$

for all $x \in \{0,1\}^n$ satisfying $\pi x \leq \bar{\pi}_0$, where $\bar{\pi}_0$ is the modified right-hand side.

In the same paper, it is further discussed that a similar construction can also be obtained for lifted cover inequalities where some of the coefficients of the left side of the cover inequality are increased to strengthen the inequality.

2.2.2 Corrected Linear Dual Functions

A natural way in which to account for the fact that linear functions are not sufficient to yield strong dual functions in the case of MILPs is to consider dual functions that consist of a linear term (as in the LP case) and a correction term accounting for the duality gap. One way to construct such a function is to consider the well-known *group relaxation*. Let B be the index set of the columns of a dual feasible basis for the LP relaxation of a PILP and denote by $N\backslash B$ the index set of the remaining columns. Consider the function F_B defined as

$$\begin{aligned} F_B(d) = \min \quad & c_B x_B + c_{N\backslash B} x_{N\backslash B} \\ \text{s.t} \quad & A_B x_B + A_{N\backslash B} x_{N\backslash B} = d \\ & x_B \in \mathbb{Z}^m, \ x_{N\backslash B} \in \mathbb{Z}^{n-m}_+. \end{aligned} \tag{2.30}$$

Substituting $x_B = A_B^{-1} d - A_B^{-1} A_{N\backslash B} x_{N\backslash B}$ in the objective function, we obtain the group relaxation (Gomory [1969])

$$\begin{aligned} F_B(d) = c_B A_B^{-1} d - \max \quad & \bar{c}_{N\backslash B} x_{N\backslash B} \\ & A_B x_B + A_{N\backslash B} x_{N\backslash B} = d \ , \\ & x_B \in \mathbb{Z}^m, \ x_{N\backslash B} \in \mathbb{Z}^{n-m}_+ \end{aligned} \tag{2.31}$$

where $\bar{c}_{N\backslash B} = (c_B A_B^{-1} A_{N\backslash B} - c_{N\backslash B})$. Here, dual feasibility of the basis A_B is required to ensure that $\bar{c}_{N\backslash B} \leq 0$.

F_B is feasible to the subadditive dual (2.6). To see this, note that F_B is subadditive since it is the sum of a linear function and the value function of a PILP. Also, we have $F_B(a^j) \leq c_B A_B^{-1} a^j - (c_B A_B^{-1} a^j - c_j) = c_j, \ j \in N\backslash B$ and $F_B(a^j) = c_B A_B^{-1} a^j = c_j, \ j \in B$. Therefore, for the PILP (1.7), $F_B(b) \leq z(b)$. Gomory [1969] further discusses sufficient conditions for F_B to be strong. Observe that $F_B(b) = z(b)$ when there exists an optimal solution to (2.31) with $x_B \geq 0$.

Another way to construct an optimal solution to the subadditive dual using a linear function with a correction term is given by Klabjan [2002].

Theorem 2.14 *(Klabjan [2002]) For a PILP in the form (1.7), and a given vector $v \in \mathbb{R}^m$, define the function F_v as*

$$F_v(d) = vd - \max\{(vA_{\mathcal{D}_v} - c_{\mathcal{D}_v})x \mid A_{\mathcal{D}_v}x \leq d, x \in \mathbb{Z}_+^{|\mathcal{D}_v|}\},$$

where $\mathcal{D}_v = \{i \in I : va^i > c_i\}$. Then, F_v is a feasible solution to the subadditive dual problem (2.6) and furthermore, if $b \in \Omega$ and $z(b) > -\infty$, there exists a $v \in \mathbb{R}^m$ such that $F_v(b) = z(b)$.

Proof. For a given v, F_v is subadditive using an argument similar to that made above for group relaxation problems. Now, consider the problem $\max\{(vA_{\mathcal{D}_v} - c_{\mathcal{D}_v})x \mid A_{\mathcal{D}_v}x \leq a^i,\ x \in \mathbb{Z}_+^{|\mathcal{D}_v|}\}$ for a given i. If $i \in I\backslash\mathcal{D}_v$, $x = 0$ is feasible. Otherwise the i^{th} unit vector is a feasible solution. Thus, for any $i \in I$, $F_v(a^i) \leq c_i$. Therefore, F_v is a feasible solution to the subadditive dual (2.6) and $F_v(b) \leq z(b)$.

Next, suppose that the original PILP is solved with Gomory's procedure (2.23) after k iterations. Let the set of generated Chvátal inequalities be represented by (π^j, π_0^j) for $j \in J = \{1, \ldots, k\}$. Let v^k and w^k be the corresponding components of the optimal dual solution with respect to the set of original constraints and the set of valid inequalities. With $x \in \{x \in \mathbb{Z}_+^{|\mathcal{D}_{v^k}|} \mid A_{\mathcal{D}_{v^k}}x = b\}$,

$$\begin{aligned}
(v^k A_{\mathcal{D}_{v^k}} - c_{\mathcal{D}_{v^k}})x &\leq -\sum_{i \in \mathcal{D}_{v^k}} \sum_{j \in J} \pi_i^j w_j^k x_i \\
&= -\sum_{j \in J} w_j^k \sum_{i \in \mathcal{D}_{v^k}} \pi_i^j x_i \\
&\leq -\sum_{j \in J} w_j^k \pi_0^j \\
&= v^k b - z(b),
\end{aligned}$$

where the first inequality follows from the dual feasibility of v^k and w^k, i.e., $v^k a^i + \sum_{j \in J} \pi_i^j w_j^k \leq$

$c_i,\ i \in \mathcal{D}_{v^k}$, and the last inequality follows from the fact that $\pi^j x \geq \pi_0^j, j \in J$, are valid inequalities for $\{A_{\mathcal{D}_{v^k}} x = b, x \in \mathbb{Z}_+^{|\mathcal{D}_{v^k}|}\}$ and $w^k \geq 0$. Rearranging, we have

$$z(b) \leq v^k b - (v^k A_{\mathcal{D}_{v^k}} - c_{\mathcal{D}_{v^k}})x \leq F_{v^k}(b). \tag{2.32}$$

Combining this result with weak duality, we get $z(b) = F_{v^k}(b)$. □

Klabjan [2002] also introduced an algorithm that finds the optimal dual function utilizing a subadditive approach from (Burdet and Johnson [1977]) together with a row generation approach that requires the enumeration of feasible solutions. Unfortunately, even for the set partitioning problems that the author reports on, this algorithm seems not to be practical.

2.2.3 Lagrangian Relaxation

Another widely used framework for generating dual problems is that of *Lagrangian duality* (Fisher [1981]). A mathematical program obtained by relaxing and subsequently penalizing the violation of a subset of the original constraints, called the *complicating* constraints, is a *Lagrangian relaxation*. Generally, this relaxation is constructed so that it is much easier to solve than the original MILP, in which case a dual problem can be constructed as follows. Suppose for a given $d \in \mathbb{R}^m$ that the inequalities defined by matrix A and right-hand side d are partitioned into two subsets defined by matrices A^1 and A^2 and right-hand sides d^1 and d^2. Furthermore, let $\mathcal{S}_{LD}(d^2) = \{x \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} : A^2 x = d^2\}$. Then, for a given penalty multiplier $v \in \mathbb{R}^{m-l}$, the corresponding Lagrangian relaxation can be formulated as

$$L(d, v) = \min_{x \in \mathcal{S}_{LD}(d^2)} cx + v(d^1 - A^1 x) \tag{2.33}$$

Assuming $z(0) = 0$ and that $x^*(d)$ is an optimal solution to the original MILP with right-hand side d, we have $L(d, v) \leq cx^*(d) + v(d^1 - A^1 x^*(d)) = cx^*(d) = z(d)\ \forall\ v \in \mathbb{R}^{m-l}$. Thus, the Lagrangian function defined by

$$L_D(d) = \max\ \{L(d, v) : v \in V\}, \tag{2.34}$$

with $V \equiv \mathbb{R}^{m-l}$, is a feasible dual function in the sense that $L_D(d) \leq z(d)\ \forall d \in \Omega$.

Note that for a given $d \in \Omega$, $L(d, v)$ is a concave, piecewise-polyhedral function. Therefore, the set V_d of extreme points of $epi(L(d, v))$ is finite. Setting $V_\Omega = \cup_{d \in \Omega} V_d$, we can rewrite $L_D(d) = \max\{L(d, v) : v \in V_\Omega\}$. It follows that if V_Ω is finite, then L_D reduces to the maximization of finitely many subadditive functions and therefore, is subadditive and feasible to the subadditive dual problem (2.6). Furthermore, in the PILP case, L_D corresponds to a Gomory function, since for a fixed v, (2.33) can be represented by a Gomory function and the maximum of finitely many Gomory functions is also a Gomory function.

L_D above is a weak dual function in general, but Blair and Jeroslow [1979] showed that it can be made strong for PILP problems by introducing a quadratic term. To show this, we first need the following proximity relation.

Theorem 2.15 *(Blair and Jeroslow [1977]) For a given PILP with $z(0) = 0$, there is a constant $\epsilon > 0$ such that*

$$|z(d_1) - z(d_2)| \leq \epsilon\ \|d_1 - d_2\|_1. \tag{2.35}$$

for all $d_1, d_2 \in \Omega$.

Let the quadratic Lagrangian relaxation be defined as

$$L(d, v, \rho) = \min_{x \in \mathbb{Z}^n_+} \{(c - vA)x + \rho \sum_{i=1}^{m} (A_i x - d_i)^2 + vd\}, \tag{2.36}$$

where $v \in \mathbb{R}^m$, $\rho \in \mathbb{R}_+$ and A_i is the i^{th} row of A.

Theorem 2.16 *(Blair and Jeroslow [1979]) For a PILP in the form (1.7), denote the quadratic Lagrangian dual function as*

$$L_D(d, v) = \max_{\rho \in \mathbb{R}_+} L(d, v, \rho). \tag{2.37}$$

Then for a given $v \in \mathbb{R}^m$, $L_D(d, v) \leq z(d)\ \forall d \in \Omega$ and furthermore, if $b \in \Omega$ and $z(b) > -\infty$, then for any $v \in \mathbb{R}^m$, $L_D(b, v) = z(b)$.

Proof. The first part follows from the fact that for any $d \in \Omega$ and $\rho \in \mathbb{R}_+$,

$$L(d, v, \rho) \leq \min_{x \in \mathcal{S}(d)} \{(c - vA)x + \rho \sum_{i=1}^{m} (A_i x - d_i)^2 + vd\} = \min_{x \in \mathcal{S}(d)} \; cx = z(d). \tag{2.38}$$

For the second part, we show that for right-hand side $b \in \Omega$ with $z(b) > -\infty$ and a given $v \in \mathbb{R}^m$, there exists $\rho(v) \in \mathbb{R}_+$ such that, $L(b, v, \rho(v)) = z(b)$. Let $\rho(v) = 1 + \epsilon + \|v\|_1$, with ϵ defined as in (2.35), assume that $\bar{x} \in \mathbb{Z}^n_+$ is an optimal solution to yield $L(b, v, \rho(v))$ and let $\bar{b} = A\bar{x}$. Then,

$$\begin{aligned}
(c - vA)\bar{x} + \rho(v) \sum_{i=1}^{m} (A_i \bar{x} - b_i)^2 + vb &= c\bar{x} + v(b - A\bar{x}) + \rho(v) \sum_{i=1}^{m} (\bar{b}_i - b_i)^2 \\
&\geq z(\bar{b}) + v(b - \bar{b}) + \rho(v)\|b - \bar{b}\|_1 \\
&\geq z(b) - \epsilon\|b - \bar{b}\|_1 - \|v\|_1\|b - \bar{b}\|_1 + \rho(v)\|b - \bar{b}\|_1 \\
&= z(b) + \|b - \bar{b}\|_1 \\
&\geq z(b)
\end{aligned} \tag{2.39}$$

by Theorem 2.15 and the fact that $\|b - \bar{b}\|_1 \leq \sum_{i=1}^{m} (b_i - \bar{b}_i)^2$. Therefore, $L(b, v, \rho(v)) \geq z(b)$ and due first part, $L_D(b, v) = L(b, v, \rho(v)) = z(b)$. □

Note that one can verify that (2.37) attains its maximum at a point x^* that is also optimal to the PILP. This is because in order to get equality in (2.39), the conditions $b = \bar{b}$ and $cx^* = z(b)$ have to be satisfied at the same time. Otherwise, $L_D(b, v) > z(b)$. In addition, it is clear that $\rho(v)$ can be replaced by any $\bar{\rho}$ such that $\bar{\rho} \geq \rho(v)$ for a given v in (2.39). In fact, if we let $\bar{v}$ be the optimal solution to the dual of the LP relaxation of PILP, then choosing $\bar{\rho} > z(b) - \bar{v}b$ is adequate, since

$$(c - \bar{v}A)\bar{x} + \bar{\rho} \sum_{i=1}^{m} (\bar{b}_i - b_i)^2 + \bar{v}b \geq \bar{\rho} + \bar{v}b > z(b). \tag{2.40}$$

Due to dual feasibility, $L(b, \bar{v}, \bar{\rho})$ is forced to have its infimum at an x^* that is also optimal to the PILP, since equality in (2.40) is attained only in that case.

2.2.4 Linear Representation of the Subadditive Dual

For bounded PILPs with $A \in \mathbb{Q}_+^{m \times n}$, the subadditive dual can be reformulated as an equivalent LP

$$\begin{array}{ll} \max & \eta(b) \\ s.t & \eta(\lambda) + \eta(\mu) \geq \eta(\lambda + \mu), \;\; 0 \leq \lambda \leq b,\, 0 \leq \mu \leq b,\, 0 \leq \lambda + \mu \leq b \\ & \eta(a^j) \leq c_j, \;\; j = 1, \dots, n \\ & \eta(0) = 0, \end{array} \tag{2.41}$$

after scaling A and b to be integer. This follows from the fact that the subadditive dual function in this case can be represented by the values it takes over the finite domain $\{\lambda \in \mathbb{Z}_+^m \mid \lambda \leq b\}$ (Gomory [1969], Johnson [1979]). The variables in the above LP represent the values of the subadditive function to be constructed at each point in this domain and the constraints ensure that the function $\eta : \{\alpha \mid \alpha \leq b\} \to \mathbb{R}$ is actually subadditive.

Lasserre [2004, 2005b] further decreases the row dimension of this LP using a discrete version of Farkas' lemma. Let $\mathbb{R}[s_1, \dots, s_m]$ be the ring of real-valued polynomials in the variables $s_i, i = 1, \dots, m$. Then, a polynomial $Q \in \mathbb{R}[s_1, \dots, s_m]$ can be written as

$$Q(s) = \sum_{\alpha \in \zeta} \lambda^\alpha s^\alpha = \sum_{\alpha \in \zeta} \lambda^\alpha s_1^{\alpha_1} \dots s_m^{\alpha_m},$$

where $\zeta \subset \mathbb{Z}_+^m$ and $\lambda^\alpha \in \mathbb{R} \; \forall \alpha \in \zeta$.

Theorem 2.17 *(Lasserre [2009]) The following two properties are equivalent:*

i. $Ax = b$ has a solution $x \in \mathbb{Z}_+^n$.

ii. The real valued polynomial $s^b - 1$ can be written as

$$s^b - 1 = \sum_{j=1}^{n} Q_j(s)(s^{a^j} - 1) \tag{2.42}$$

for some real-valued polynomials $Q_j \subset \mathbb{R}[s_1, \ldots, s_m], j = 1, \ldots, n$, *all with nonnegative coefficients.*

Proof. (1)→(2). Let $x \in \mathcal{S}$. Writing

$$s^b - 1 = s^{a^1 x_1} - 1 + s^{a^1 x_1}(s^{a^2 x_2} - 1) + \cdots + s^{\sum_{j=1}^{n-1} a^j x_j}(s^{a^n x_n} - 1)$$

with

$$s^{a^j x_j} - 1 = (s^{a^j} - 1)\left[1 + s^{a^j} + \cdots + s^{a^j(x_j - 1)}\right],\ j = 1, \ldots, n,$$

we obtain

$$Q_j(s) = s^{\sum_{k=1}^{j-1} a^k x_k}\left[1 + s^{a^j} + \cdots + s^{a^j(x_j - 1)}\right],\ j = 1, \ldots, n. \tag{2.43}$$

(2)→(1). Let $q \in \mathbb{R}^k_+$ be the vector of nonnegative coefficients of all polynomials $Q_j,\ j = 1, \ldots n$, and $M \in \mathbb{R}^{p \times k}$ be such that the set of constraints defining the polyhedron $\Theta = \{q \mid Mq = \tau,\ q \geq 0\} \neq \emptyset$ equalizes the respective coefficients of the polynomials $s^b - 1$ and $\sum_{j=1}^n Q_j(s)(s^{a^j} - 1)$. It is easy to show that each $Q_j,\ j = 1, \ldots n$, may be restricted to contain only monomials $\{s^\alpha : \alpha \leq b - a^j,\ \ \alpha \in \mathbb{Z}^m_+\}$ and therefore

$$\begin{aligned} p &= \prod_{i=1}^m (b_i + 1) \\ k &= \sum_{j=1}^n k_j \quad \text{with} \quad k_j = \prod_{i=1}^m (b_i - a^j_i + 1), \quad j = 1, \ldots, n. \end{aligned}$$

In other words, p is the number of monomials y^α with $\alpha \leq b$ and k_j is the number of monomials y^α with $\alpha - a^j \leq b$. With this construction, it is not hard to see that M is totally unimodular and each extreme point of Θ, if it exists, is integral, since τ is also integral.

Next, recall that $\mathbf{1}^{k_j} = (1, \ldots, 1) \in \mathbb{R}^{k_j},\ j = 1, \ldots, n$, and let $K \in \mathbb{Z}^{n \times k}_+$ be the n-block diagonal matrix, whose each diagonal block is a row vector $\mathbf{1}^{k_j}$, that is,

$$K = \begin{bmatrix} \mathbf{1}^{k_1} & 0 & \dots & 0 \\ 0 & \mathbf{1}^{k_2} & 0 & \dots \\ \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & \mathbf{1}^{k_n} \end{bmatrix}.$$

Now, let $Q_j, j = 1, \dots, n$, be the set of polynomials satisfying (2.42). Then, $\Theta \neq \emptyset$ and there exists an integral $\bar{q} \in \Theta$. If we denote by $\bar{Q}_j, j = 1, \dots, n$, the corresponding monomials $\bar{q}$ represents and take the derivative of both sides with respect to $s_i, i = 1, \dots, m$, at $(1, \dots, 1)$, we get

$$b_i = \sum_{j=1}^{n} \bar{Q}_j(1, \dots, 1) a_i^j = \sum_{j=1}^{n} a_i^j (K\bar{q})_j, \quad , i = 1, \dots, m.$$

Observe that setting $x = K\bar{q}$ completes the proof. □

The converse of the last part of the proof is also valid, i.e., for any $x \in \mathcal{S}, x = Kq$ for some $q \in \Theta$. As a consequence, we have the following corollary.

Corollary 2.18 *(Lasserre [2004]) For a PILP in the form (1.7) with $A \in \mathbb{Z}_+^{m \times n}$, let K, M, τ be defined as before. Then, $z(b) = \min\{cKq \mid Mq = \tau\ , \ \ q \geq 0\}$. Moreover, if q^* is an optimal solution, then $x^* = Kq^*$ is an optimal solution to the PILP.*

Lasserre further shows that the LP dual of the problem in the first part of Corollary 2.18 can be reduced to a subadditive formulation that is also dual to PILP. Compared to (2.41), the number of variables is the same, however, this one has $\mathcal{O}(np)$ constraints, whereas (2.41) has $\mathcal{O}(p^2)$ constraints.

2.2.5 Branch And Cut

The most common technique for solving MILPs in practice today is the branch-and-cut algorithm. Developing a procedure for obtaining a dual function as a by-product of this procedure is of great importance if duality is to be made computationally useful. Here we discuss "vanilla" branch-and-cut, in which branching is done only by restricting variable bounds and no standard computational enhancements, such as preprocessing, are used. Such an algorithm works by attempting to solve the subproblem of each branch-and-cut tree node utilizing a cutting plane method, as described in Section 2.2.1. If the subadditive characterization or a functional encoding of the right-hand side dependency is available for each cut, then we can obtain a dual function for the corresponding subproblem. Below, we show how this dual information can be gathered together to yield a feasible dual function for the original problem.

Assume that the MILP (1.7) has a finite optimum and has been solved to optimality with a branch-and-cut algorithm. Let T be the set of leaf nodes of the tree and let $\nu(t)$ be the number of cuts generated so far on the path from the root node to node $t \in T$ (including the ones generated at t). To obtain a bound for this node, we solve the LP relaxation of the following problem

$$\begin{aligned} z^t(b) = \min \quad & cx \\ \text{s.t.} \quad & x \in \mathcal{S}_t(b), \end{aligned} \tag{2.44}$$

where the feasible region $\mathcal{S}_t(b) = \{x \in \mathbb{Z}^r \times \mathbb{R}^{n-r} \mid Ax = b, x \geq l^t, -x \geq -u^t, \Pi^t x \geq \Pi_0^t\}$ and $u^t,\ l^t \in \mathbb{Z}_+^n$ are the branching bounds applied to the integer variables, $\Pi^t \in \mathbb{R}^{\nu(t)\times n}$ and $\Pi_0^t \in \mathbb{R}^{\nu(t)}$.

For each cut k, $k = 1, \ldots, \nu(t)$, suppose that the subadditive representation F_k^t is known and let the function σ_k^t be defined for (2.44) as in Section 2.2.1, considering also the branching bounds. For each feasibly pruned node $t \in T$, let $(v^t, \underline{v}^t, \overline{v}^t, w^t)$ be the corresponding dual feasible solution used to obtain the bound that allowed the pruning of node t. Note that such a solution is always available if the LP relaxations are solved using a dual simplex algorithm. For each infeasibly pruned node $t \in T$, let $(v^t, \underline{v}^t, \overline{v}^t, w^t)$ be a corresponding dual feasible solution with $v^t b + \underline{v}^t l^t - \overline{v}^t u^t + w^t \Pi_0^t \geq z(b)$ that can be obtained from the dual solution of the parent of node t and a dual ray that makes the dual problem unbounded.

Theorem 2.19 *If $b \in \Omega$ and $z(b) > -\infty$, then the function*

$$F_{BC}(d) = \min_{t \in T}\{v^t d + \underline{v}^t l^t - \overline{v}^t u^t + \sum_{k=1}^{\nu(t)} w_k^t F_k^t(\sigma_k^t(d))\} \tag{2.45}$$

is an optimal solution to the dual (1.20).

Proof. The proof follows the outline of Wolsey [1981]'s proof for validating an optimal dual function for the branch-and-bound algorithm. Because of the way branch-and-cut algorithm partitions $\mathcal{S}$, we are guaranteed that for any $d \in \Omega$ and $\hat{x} \in \mathcal{S}(d)$, there must exist a leaf node $t \in T$ such that $\hat{x} \in \mathcal{S}_t(d)$. Then, from LP duality,

$$c_j\hat{x}_j \geq v^t a^j \hat{x}_j + \underline{v}_j^t \hat{x}_j - \overline{v}_j^t \hat{x}_j + w^t \Pi_j^t \hat{x}_j \qquad j = 1, \ldots, n, \tag{2.46}$$

where Π_j^t is the j^{th} column of Π^t. Adding the above inequalities over all columns, we get

$$\begin{aligned} c\hat{x} &\geq v^t A\hat{x} + \underline{v}^t \hat{x} - \overline{v}^t \hat{x} + w^t \Pi^t \hat{x} \\ &\geq v^t d + \underline{v}^t l^t - \overline{v}^t u^t + \sum_{k=1}^{\nu(t)} w_k^t F_k^t(\sigma_k^t(d)) \\ &\geq F_{BC}(d). \end{aligned} \tag{2.47}$$

Now assume that x^* is an optimal solution to MILP with right-hand side b. In this case, we know that for some node t^*, $z(b) = cx^* = z^{t^*}(b)$ and we also have that $z^t(b) \geq z^{t^*}(b)$ for all $t \in T$. Therefore, $F_{BC}(b) = z(b)$. □

Unfortunately, (2.45) is not subadditive due to the the constant term resulting from the bounds imposed by branching and hence is not feasible for the subadditive dual (2.6). One can, however, obtain a subadditive dual function in the case where the original MILP has explicit upper and lower bounds on all variables by including these bounds as part of the right-hand side. Suppose that

$$\bar{z}(\bar{b}) = \min\{cx \mid \bar{A}x \geq \bar{b},\ x \in \mathbb{Z}^r \times \mathbb{R}^{n-r}\} \tag{2.48}$$

with $\bar{A} = [\,A \;\; I \; -I\,]'$ and $\bar{b} = [\,b \;\; l \; -u\,]$ where l and u are the lower and upper bounds pre-defined on the variables. With this construction, at each node $t \in T$, we solve the LP relaxation of the following subproblem

$$\begin{aligned} \bar{z}_t(\bar{b}^t) = \min \quad & cx \\ \text{s.t.} \quad \bar{A}x &\geq \bar{b}^t \\ \Pi^t x &\geq \Pi_0^t \\ x &\in \mathbb{Z}^r \times \mathbb{R}^{n-r} \end{aligned} \tag{2.49}$$

with $\bar{b}^t = [\,b \;\; l^t \;\; -u^t\,]$.

Theorem 2.20 *If $\mathcal{S}(\bar{b}) \neq \emptyset$ and $\bar{z}(\bar{b}) > -\infty$, then the function*

$$F_{BCS}(d) = \max_{t \in T}\{\bar{v}^t d + \sum_{k=1}^{\nu(t)} \bar{w}_k^t F_k^t(\sigma_k^t(d))\}, \quad d \in \mathbb{R}^{m+2n} \tag{2.50}$$

is feasible to the subadditive dual problem (2.6) of the MILP (2.48).

Proof. For any $t \in T$, LP duality yields

$$c_j \geq \bar{v}^t \bar{a}^j + \sum_{k=1}^{\nu(t)} \bar{w}_k^t \Pi_j^t \qquad j = 1, \ldots, n.$$

Therefore, it is clear that $c_j \geq F_{BCS}(\bar{a}^j)$ if $j \in I$ and likewise, $c_j \geq \bar{F}_{BCS}(\bar{a}^j)$ when $j \in C$. In addition, since $F_k^t \in \Gamma^{m+2n+k-1},\ k = 1, \ldots, \nu(t)$, $F_{BCS} \in \Gamma^{m+2n}$. □

Note that in this case, the dual function may not be strong. As in Theorem 2.13, it is not strictly necessary to have a subadditive representation of each cut in order to apply the results of this section. They remain valid as long as a functional dependency of each cut on the right-hand side is known (see Section 2.2.1).

The theorems above remain valid when the original right-hand side dependency of each cut is known (see Section 2.2.1). However, in the absence of this information, at least for bounded MILPs, we can derive a weak dual function by considering the bounds on each variable. That is, we can either modify the right-hand side of each cut to the smallest possible left-hand-side value achievable by setting variables at their respective bounds or adjust the level of the LP dual objective to be a lower bound for any feasible primal solution, again via the variable bounds. To see this, for each node $t \in T$ let

$$\hat{h}^t_k = \sum_{j=1}^{n} \Pi^t_{kj} \hat{x}_j \qquad \text{with} \qquad \hat{x}_j = \begin{cases} l^t_j & \text{if } \Pi^t_{kj} \geq 0 \\ u^t_j & \text{otherwise} \end{cases} \quad , \quad k = 1, \ldots, \nu(t),$$

where Π^t_{kj} is the k^{th} entry of column Π^t_j. Furthermore, define

$$\tilde{h}^t = \sum_{j=1}^{n} \Pi^t_j \tilde{x}_j \qquad \text{with} \qquad \tilde{x}_j = \begin{cases} l^t_j & \text{if } w^t \Pi^t_j \geq 0 \\ u^t_j & \text{otherwise} \end{cases} \quad .$$

Theorem 2.21 *If $b \in \Omega$ and $z(b) > -\infty$, then the function*

$$F_{BCV}(d) = \min_{t \in T} \{ v^t d + \underline{v}^t l^t - \overline{v}^t u^t + \max\{w^t \tilde{h}^t,\ w^t \hat{h}^t\}\} \tag{2.51}$$

is a feasible solution to the dual (1.20).

Proof. With the same argument as in the proof of Theorem 2.19, for any right-hand side d and $x \in \mathcal{S}(d)$, let $t \in T$ be such that $x \in S_t(d)$. Then from LP duality,

$$\begin{aligned} cx &\geq v^t A x + \underline{v}^t x - \overline{v}^t x + w^t \sum_{j=1}^{n} \Pi^t_j x_j \\ &\geq v^t d + \underline{v}^t l^t - \overline{v}^t u^t + \max\{w^t \tilde{h}^t,\ w^t \hat{h}^t\} \\ &\geq F_{BCV}(d). \end{aligned} \tag{2.52}$$

□

So far, we have reviewed duality for mixed integer linear programs and extended the possible ways of obtaining dual functions to the realm of branch-and-cut algorithm. Our next step is to discuss how to use this dual information to derive practical techniques both for approximating the value function (Chapter 3) and for the ability to perform sensitivity analyses and to warm start solution processes (Chapter 4).

Chapter 3

The Value Function of a Mixed Integer Linear Program

This chapter concerns the structure and approximation of the value function of an MILP with the goal of using these approximations as a substitute for the value function in large-scale algorithms. Herein, we first overview the previous results on the structure of the value function. Next, we look more closely at the structure of the value function of an MILP with a single constraint and non-negativity constraints on all variables. We show for this case that the value function is uniquely determined by a finite number of break points and at most two slopes, derive conditions for the value function to be continuous and suggest a method for systematically extending the value function from a specified neighborhood of the origin to the entire real line. Although we focus here in particular on this specific case, we point out that a number of these results hold in more general settings. Then we discuss both upper and lower approximations of the value function of a MILP through the value functions of single-constraint relaxations, dual functions and restrictions. Finally, we outline methods for embedding these approximation techniques into solution methods for specific applications: stochastic integer programming and bilevel integer programming problems.

3.1 Overview

The value function itself is the most useful dual function we can obtain for studying the effect of perturbations of the right-hand side vector, since it provides an exact solution value for any right-hand side vector. Unfortunately, it is unlikely that there exist effective methods for constructing the value function for general MILPs. For PILPs, Blair and Jeroslow [1982] showed that a procedure similar to Gomory's cutting plane procedure can be used to construct the value function in a finite number of steps. The representation from this procedure may have exponential size. However, they were able to characterize the class of functions to which value functions belong, namely, *Gomory functions*, a subset of a more general class called *Chvátal functions*.

Definition 3.1 *Chvátal functions are the smallest set of functions $\mathscr{C}^m$ such that*

i. If $h \in \mathcal{L}^m$, then $h \in \mathscr{C}^m$.

ii. If $h_1, h_2 \in \mathscr{C}^m$ and $\alpha, \beta \in \mathbb{Q}_+$, then $\alpha h_1 + \beta h_2 \in \mathscr{C}^m$.

iii. If $h \in \mathscr{C}^m$, then $\lceil h \rceil \in \mathscr{C}^m$.

Gomory functions are the smallest set of functions $\mathscr{G}^m \subseteq \mathscr{C}^m$ with the additional property that

4. If $h_1, h_2 \in \mathscr{G}^m$, then $\max\{h_1, h_2\} \in \mathscr{G}^m$.

The relationship between $\mathscr{C}^m$ and $\mathscr{G}^m$ is evident from the following theorem.

Theorem 3.1 *(Blair and Jeroslow [1982]) Every Gomory function can be written as the maximum of finitely many Chvátal functions, that is, if $g \in \mathscr{G}^m$, then there exist $h_i \in \mathscr{C}^m$ for $i = 1, \ldots, k$ such that*

$$g = \max\{h_1, \ldots, h_k\}. \tag{3.1}$$

This theorem also makes evident the relationship between $\mathscr{G}^m$ and the property of subadditivity. Note that if h_1, h_2 are subadditive and $\alpha, \beta \in \mathbb{Q}_+$, then it is easy to show that the functions $\alpha h_1 + \beta h_2$ and $\lceil h_1 \rceil$ are both subadditive. Consequently, one can show that Chvátal functions are subadditive by induction on the rank of functions (i.e., the number of operations of the type specified in Definition 3.1 needed to derive a given Chvátal function from the base class $\mathcal{L}^m$). Since

$\max\{h_1, h_2\}$ is subadditive, Gomory functions are also subadditive. As a result of subadditivity, both Chvátal and Gomory functions can be used for generating valid inequalities. The following lemma, needed for the proof of Theorem 3.3 shows that for PILPs, Chvátal functions can be used to obtain a description of the convex hull of solutions.

Lemma 3.2 *(Schrijver [1980]) The subadditive functions in Theorem 2.11 can be taken to be Chvátal functions.*

The above lemma then allows us to characterize the value function of a PILP for which $z(0) = 0$.

Theorem 3.3 *(Blair and Jeroslow [1982]) For a PILP in the form (1.7), if $z(0) = 0$, then there is a $g \in \mathscr{G}^m$ such that $g(d) = z(d)$ for all $d \in \Omega$.*

Proof. Consider the parameterized family of PILPs $\min\{cx \mid x \in conv(\mathcal{S}(d))\}\ \forall d \in \Omega$, where $\mathrm{conv}(\mathcal{S}(\mathrm{d}))$ is represented by the finite set of Chvátal functions whose existence is guaranteed by Lemma 3.2. Applying LP duality, we get $g(d) = z(d) = \max\{\sum_{i=1}^{k} v_i F_i(d) \mid v \in V\}$ where V is the finite set of dual basic feasible solutions. Then the proof is complete by Theorem 3.1. □

Example 6 The value function of problem (2.1) with all variables assumed to be integer can be written as $z(d) = \frac{3}{2}\max\{\lceil\frac{2d}{3}\rceil,\ \lceil d\rceil\} - d\ \forall d \in \Omega$, which is a Gomory function (see Figure 2.4)□

For PILPs, it is also worth mentioning that there always exists an optimal solution to the subadditive dual problem (2.6) that is a Chvátal function.

Theorem 3.4 *(Blair and Jeroslow [1982]) For a PILP in the form (1.7), if $b \in \Omega$ and $z(b) > -\infty$, then there exists $h \in \mathscr{C}^m$ that is optimal to the subadditive dual (2.6).*

Proof. Note from the proof of Theorem 2.6 that either the value function itself, or an extension of the value function is a feasible solution to the subadditive dual. Denote this function as z_e. From Theorem 3.3, we know that there is a $g \in \mathscr{G}^m$ with $g(d) = z_e(d)$ for all $d \in \Omega$ and hence, feasible to the subadditive dual (2.6). By Theorem 3.1, g is the maximum of finitely many Chvátal functions, $h_1, \ldots, h_k$. For right-hand side b, since $z_e(b) = \max\{h_1(b), \ldots h_k(b)\}$, there exists

$l \in \{1, \ldots, k\}$ with $z(b) = z_e(b) = h_l(b)$. Then h_l is an optimal solution to the subadditive dual (2.6) since it is subadditive, and $h_l(a^j) \leq g(a^j) \leq c_j$ for all $j \in I$. □

Note that F_G, obtained by (2.28) through cutting-plane method, is also a Chvátal function and hence, can be seen as an alternative proof for Theorem 3.4.

Using a result similar to Corollary 2.7 above, Blair and Jeroslow [1982] introduced the more general concept of a *consistency tester* to detect the infeasibility of the problem for any right-hand side. They showed that for a given PILP, there is a $g \in \mathscr{G}^m$ such that for every $d \in \mathbb{R}^m$, $g(d) \leq 0$ if and only if $d \in \Omega$. Using the consistency tester concept, we can state a converse of Theorem 3.3. That is, for Gomory functions g_1, g_2, there exist $\bar{A}, \bar{c}$ such that $g_1(d) = \min\{\bar{c}x \mid \bar{A}x = d,\ x \geq 0$ and integral$\}$ for all d with $g_2(d) \leq 0$. In this sense, there is one-to-one correspondence between PILP instances and Gomory functions.

Jeroslow Formula. For MILPs, neither Theorem 3.3 nor its converse holds. However, Blair and Jeroslow [1984] argue that the value function z can still be represented by a Gomory function if $c_j = 0\ \forall j \in C$ or can be written as a minimum of finitely many Gomory functions. A deeper result is contained in the subsequent work of Blair [1995], who showed that the value function of an MILP can be written as a *Jeroslow formula*, consisting of a Gomory function and a correction term. Here, rather than the formula itself (see Blair and Jeroslow [1984], Blair [1995] for details), we present a simplified version to illustrate its structure.

For a given $d \in \Omega$, let the set $\mathscr{E}$ consist of the index sets of dual feasible bases of the linear program

$$\min\{c_C x_C \ : \ A_C x_C = d, x \geq 0\}. \tag{3.2}$$

By the rationality of A, we can choose $M \in \mathbb{Z}_+$ such that $MA_E^{-1}a^j \in \mathbb{Z}^m$ for all for all $E \in \mathscr{E}$, $j \in I$. For $E \in \mathcal{E}$, let v_E be the corresponding basic feasible solution to the dual of

$$\min\{\frac{1}{M}c_C x_C \ : \ \frac{1}{M}A_C x_C = d, x \geq 0\}, \tag{3.3}$$

which is a scaled version of (3.2). Finally, for a right-hand side d and $E \in \mathcal{E}$, let $\lfloor d \rfloor_E =$

$A_E\lfloor A_E^{-1}d\rfloor$.

Theorem 3.5 *(Blair [1995]) For the MILP (1.7), there is a $g \in \mathscr{G}^m$ such that*

$$z(d) = \min_{E\in\mathscr{E}} g(\lfloor d\rfloor_E) + v_E(d - \lfloor d\rfloor_E) \tag{3.4}$$

for all $d \in \Omega$.

Proof. Assume that c_C and A_C are scaled as in (3.3) and consider the PILP instance

$$\begin{aligned} z_{JF}(\phi) = \min \quad & cx + z(\varphi)y \\ \text{s.t} \quad & Ax + \varphi y = \phi \\ & x \in \mathbb{Z}^n_+, y \in \mathbb{Z}_+ \ , \end{aligned} \tag{3.5}$$

where $\varphi = -\sum_{j\in C} a^j$. Then we have the following:

i. For a given $E \in \mathscr{E}$ and $d \in \mathbb{R}^m$, (3.5) is feasible for $\phi = \lfloor d\rfloor_E$. To see this, observe that if $\lfloor A_E d\rfloor \geq 0$, then $x_E = \lfloor A_E^{-1}d\rfloor,\ x_{N\setminus E} = 0,\ y = 0$ is a feasible solution. Otherwise, there exists $\Delta \in \mathbb{Z}_+$ such that $x_E = (\lfloor A_E^{-1}d\rfloor + \Delta A_E^{-1}\sum_{j\in E} a^j) \in Z^m_+$, since $A_E^{-1}\sum_{j\in E} a^j = \mathbf{1}^m$. Therefore, together with x_E, $x_I = 0$, $x_j = \Delta$ for $j \in C\setminus E$ and $y = \Delta$ is a feasible solution.

ii. For a giveb $E \in \mathscr{E}$ and $d \in \mathbb{R}^m$, $z_{JF}(\lfloor d\rfloor_E) \geq z(\lfloor d\rfloor_E)$. To see this, assume that $z(\varphi) = cx_1$ and $z_{JF}(\lfloor d\rfloor_E) = cx_2 + z(\varphi)\hat{y} = c(x_2 + x_1\hat{y})$. But then, clearly, $z_{JF}(\lfloor d\rfloor_E) \geq z(\lfloor d\rfloor_E)$ since $(x_2 + x_1\hat{y}) \in \mathcal{S}(\lfloor d\rfloor_E)$.

Now, we know from Theorem 3.3 that there is a $g \in \mathscr{G}^m$ with $g(\lfloor d\rfloor_E) = z_{JF}(\lfloor d\rfloor_E)$ for all $d \in \mathbb{R}^m,\ E \in \mathscr{E}$. Let $d \in \Omega$ be given and $\bar{x} \in \mathcal{S}(d)$. By LP duality, there is an $E \in \mathscr{E}$ with

$c_E A_E^{-1} A_C \bar{x}_C \leq c_C \bar{x}_C$. Noting that $\lfloor d \rfloor_E = \lfloor A\bar{x} \rfloor_E = A_I \bar{x}_I + \lfloor A_C \bar{x}_C \rfloor_E$, we have

$$\begin{aligned} g(\lfloor d \rfloor_E) + v_E(d - \lfloor d \rfloor_E) &= z_{JF}(A_I \bar{x}_I + \lfloor A_C \bar{x}_C \rfloor_E) + v_E(d - \lfloor d \rfloor_E) \\ &\leq c_I \bar{x}_I + c_E A_E^{-1} \lfloor A_C \bar{x}_C \rfloor_E + v_E(d - \lfloor d \rfloor_E) \\ &= c_I \bar{x}_I + c_E A_E^{-1} A_C \bar{x}_C \\ &\leq c\bar{x}, \end{aligned}$$

where the first inequality follows from the fact that $x_I = \bar{x}_I$, $x_j = 0$ for $j \in C \backslash E$, $x_E = A_E^{-1} \lfloor A_C \bar{x}_C \rfloor_E$, and $y = 0$ is a feasible solution to (3.5) with right-hand side $A_I \bar{x}_I + \lfloor A_C \bar{x}_C \rfloor_E$ and the last equality follows from the fact that

$$v_E(d - \lfloor d \rfloor_E) = v_E(A_C \bar{x}_C - \lfloor A_C \bar{x}_C \rfloor_E) = c_E A_E^{-1}(A_C \bar{x}_C - \lfloor A_C \bar{x}_C \rfloor_E).$$

On the other hand, for a given $E \in \mathscr{E}$,

$$g(\lfloor d \rfloor_E) + v_E(d - \lfloor d \rfloor_E) = z_{JF}(\lfloor d \rfloor_E) + v_E(d - \lfloor d \rfloor_E) \geq z(\lfloor d \rfloor_E) + z(d - \lfloor d \rfloor_E) \geq z(d).$$

by the subadditivity of z. □

Example 7 Consider the MILP (2.1). With $M = 2$, the set of index sets of dual feasible bases of $\min\{x_3 + \frac{1}{2}x_4 \mid \frac{1}{2}x_3 - \frac{1}{2}x_4 = d,\ x_3, x_4 \geq 0\}$ is $\mathscr{E} = \{\{3\}, \{4\}\}$. Furthermore, $v_{\{3\}} = 2$ and $v_{\{4\}} = -1$. Since $\varphi = \frac{1}{2} - \frac{1}{2} = 0$ and $z(0) = 0$, (3.5) reduces to $z_{JF}(\phi) = \{\frac{1}{2}x_1 + x_3 + \frac{1}{2}x_4 \mid x_1 - \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_4 = \phi, x_i \in \mathbb{Z}_+, i = 1, \ldots, 4\}$. The value function of this problem is the same as z in Example 6. Thus, $g(d) = \frac{3}{2}\max\{\lceil \frac{2d}{3} \rceil,\ \lceil d \rceil\} - d$ solves this problem. Then the value function (see Figure 2.1) of (2.1) is

$$\min\left\{\frac{3}{2}\max\left\{\left\lceil \frac{\lfloor 2d \rfloor}{3} \right\rceil, \left\lceil \frac{\lfloor 2d \rfloor}{2} \right\rceil\right\} + \frac{3\lceil 2d \rceil}{2} + 2d,\ \frac{3}{2}\max\left\{\left\lceil \frac{\lceil 2d \rceil}{3} \right\rceil, \left\lceil \frac{\lceil 2d \rceil}{2} \right\rceil\right\} - d\right\}.$$

□

Generating Functions. Lasserre [2005a,b] recently introduced a different method for constructing the value function of PILPs that utilizes generating functions. This methodology does not fit well into a traditional duality framework, but nevertheless gives some perspective about the role of basic feasible solutions of the LP relaxation in determining the optimal solution of a PILP.

Theorem 3.6 *(Lasserre [2009]) For a PILP in the form (1.7) with $A \in \mathbb{Z}^{m\times n}$, define*

$$z(d,c) = \min_{x\in\mathcal{S}(d)} cx, \tag{3.6}$$

and let the corresponding summation function be

$$\hat{z}(d,c) = \sum_{x\in\mathcal{S}(d)} e^{cx}\ \forall d \in \mathbb{Z}^m. \tag{3.7}$$

Then the relationship between z and $\hat{z}$ is

$$e^{z(d,c)} = \lim_{q\to-\infty} \hat{z}(d,qc)^{1/q} \quad \textit{or equivalently,} \quad z(d,c) = \lim_{q\to-\infty} \frac{1}{q}\ln\hat{z}(d,qc). \tag{3.8}$$

In order to get a closed form representation of $\hat{z}$, we can solve the two sided $\mathbb{Z}$ - transform $\hat{F} : \mathbb{C}^m \to \mathbb{C}$ defined by

$$\hat{F}(s,c) = \sum_{d\in\mathbb{Z}^m} s^{-d}\hat{z}(d,c) \tag{3.9}$$

with $s^d = s_1^{d_1}\dots s_m^{d_m}$ for $d \in \mathbb{Z}^m$. Substituting $\hat{z}$ in this formula, we get

$$\begin{aligned}
\hat{F}(s,c) &= \sum_{d\in\mathbb{Z}^m}\sum_{x\in\mathcal{S}(d)} s^{-d}e^{cx} \\
&= \sum_{x\in\mathbb{Z}^n_+} e^{cx}\sum_{d=Ax} s^{-d} \\
&= \sum_{x\in\mathbb{Z}^n_+} e^{cx}s^{-Ax} \\
&= \prod_{j=1}^{n}\frac{1}{1-e^{c_j}s^{-a^j}}, \qquad (3.10)
\end{aligned}$$

where the last equality is obtained by applying Barvinok [1993]'s short form equation for summation problems over a domain of all non-negative integral points. The formula (3.10) is well-defined if $|s^{a^j}| > e^{c_j},\ j = 1, \ldots, n$ and the function $\hat{z}$ is then obtained by solving the inverse problem

$$\hat{z}(d, c) = \frac{1}{(2i\pi)^m} \int_{|s|=\gamma} \hat{F}(s, c) s^{d-1^m} ds = \frac{1}{(2i\pi)^m} \int_{|s_1|=\gamma_1} \cdots \int_{|s_m|=\gamma_m} \hat{F}(s, c) s^{d-\mathbf{1}^m} ds, \tag{3.11}$$

where γ is a vector satisfying $\gamma^{a^j} > e^{c_j}\ \ j = 1, \ldots, n$ and $\mathbf{1}^m = (1, \ldots, 1) \in \mathbb{R}^m$.

Although it is possible to solve (3.11) directly by Cauchy residue techniques, the complex poles make it difficult. One alternative is to apply Brion and Vergne's (see Brion and Vergne [1997], Lasserre [2009] for details) lattice points counting formula in a polyhedron to get the reduced form, which, for each $d \in \mathbb{R}^m$, is composed of the optimal solution value of the LP relaxation and a correction term. The correction term is the minimum of the sum of the reduced costs of certain nonbasic variables over all basic feasible solutions, obtained by the degree sum of certain real-valued univariate polynomials. Another approach using generating functions is to apply Barvinok [1994]'s algorithm for counting lattice points in a polyhedron of fixed dimension to a specially constructed polyhedron that includes for any right-hand side the corresponding minimal test set (see Loera et al. [2004a,b] for details).

3.2 The Value Function of an MILP with a Single Constraint

In what follows, we consider an MILP with one equality constraint, i.e., we set $m = 1$, which reduces to the problem of determining

$$z_P = \min_{x \in \mathcal{S}} cx, \tag{P}$$

where $\mathcal{S} = \{x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \mid ax = b\}$, $a \in \mathbb{Q}^n$. In this case, the value function is

$$z(d) = \min_{x \in \mathcal{S}(d)} cx, \tag{3.12}$$

where for a given $d \in \mathbb{R}$, $\mathcal{S}(d) = \{x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \mid ax = d\}$.

To simplify the presentation as much as possible, we make a number of assumptions. First, we assume that $z(0) = 0$, since otherwise $z(d) = -\infty$ for all $d \in \mathbb{R}$ with $\mathcal{S}(d) \neq \emptyset$. Although our results remain valid when either $a \in \mathbb{Q}^n_+$ or $a \in \mathbb{Q}^n_-$, we assume that both $N^+ = \{i \in N \mid a_i > 0\}$ and $N^- = \{i \in N \mid a_i < 0\}$ are nonempty since otherwise, $z(d) = \infty\ \forall d \in \mathbb{R}_- \backslash \{0\}$ in the first case and $\forall d \in \mathbb{R}_+ \backslash \{0\}$ in the latter case. Finally, we assume $r < n$, that is, at least one of $C^+ = \{i \in C \mid a_i > 0\}$ and $C^- = \{i \in C \mid a_i < 0\}$ is nonempty, to ensure that $S(d) \neq \emptyset$ for all $d \in \mathbb{R}$. Taking all of these assumptions together, we have that $-\infty < z(d) < \infty\ \forall d \in \mathbb{R}$. Note that all of these assumptions can be easily relaxed. In particular, the last assumption, which may appear restrictive, is made without loss of generality, since by Theorem 2.5, the value function of a PILP with a single constraint can always be extended to that of a corresponding MILP with a single constraint in such a way that the two functions agree for all right-hand sides $d \in \mathbb{R}$ for which the original PILP is feasible.

Consequently, for a fixed right-hand side $b \in \mathbb{R}$, the *subadditive dual* of the MILP (P) reduces to

$$\begin{array}{ll} \max & F(b) \\ & F(a_j) \leq c_j \quad j \in I, \\ & \bar{F}(a_j) \leq c_j \quad j \in C, \\ & F(0) = 0 \text{ and } F \text{ is subadditive,} \end{array} \quad . \tag{D}$$

where, as we discussed earlier, the function $\bar{F}$, is the upper d-directional derivative of F.

3.2.1 Piecewise-Linear Approximations

Obviously, one possibility for approximating the value function is to consider the value function of its LP relaxation. In this case, the value function of the LP relaxation, which we shall denote by F_L, has a convenient closed form. Let

$$\eta = \min\{\frac{c_i}{a_i} \mid i \in N^+\} \text{ and } \zeta = \max\{\frac{c_i}{a_i} \mid i \in N^-\}. \tag{3.13}$$

Then we have

$$F_L(d) = \max\{ud \mid \zeta \le u \le \eta, u \in \mathbb{R}\} = \begin{cases} \eta d & \text{if } d > 0 \\ 0 & \text{if } d = 0 \\ \zeta d & \text{if } d < 0 \,. \end{cases} \tag{3.14}$$

Note that we must have $\eta \ge \zeta$, since otherwise the dual problem of the LP relaxation of (P) is infeasible, which in turn, is a contradiction to our initial assumptions that $a \in \mathbb{Q}^n$ and $z(0) = 0$. Similarly, we can get an upper bound on the value function by considering only the continuous variables. Let

$$\eta^C = \min\{\frac{c_i}{a_i} \mid i \in C^+\} \text{ and } \zeta^C = \max\{\frac{c_i}{a_i} \mid i \in C^-\}. \tag{3.15}$$

By convention, if $C^+ = \emptyset$, then we set $\eta^C = \infty$. Similarly, if $C^- = \emptyset$, then we set $\zeta^C = -\infty$. The function F_U, defined by

$$F_U(d) = \min\{c_C x_c \mid a_C x_C = d, x_i \ge 0 \;\; \forall i \in C\} = \begin{cases} \eta^C d & \text{if } d > 0 \\ 0 & \text{if } d = 0 \\ \zeta^C d & \text{if } d < 0 \,, \end{cases} \tag{3.16}$$

is then an upper bound on z. This follows from the fact that for a given feasible right-hand side, any optimal solution to the LP (3.16) can be extended to a feasible solution for the original MILP (P) by fixing all the integer variables to 0. In addition, $z(d) = F_U(d) = F_L(d)\ \forall d \in \mathbb{R}_+$ if and only if $\eta = \eta^C$ and similarly, $z(d) = F_U(d) = F_L(d)\ \forall d \in \mathbb{R}_-$ if and only if $\zeta = \zeta^C$.

Before proceeding, we note that the above bounding functions are only special cases of the general lower and upper bounding approximations we discuss later in Sections 3.3.2 and 3.3.3. Although we can extract closer and tighter approximations, the ones given here are sufficient for the arguments we make in the rest of this chapter for the structure of the value function (3.12).

Example 8 Consider the following MILP instance with a fixed right-hand side b

$$\begin{aligned} \min \quad & 3x_1 + \tfrac{7}{2}x_2 + 3x_3 + 6x_4 + 7x_5 + 5x_6 \\ \text{s.t} \quad & 6x_1 + 5x_2 - 4x_3 + 2x_4 - 7x_5 + x_6 = b \quad \text{and} \\ & x_1, x_2, x_3 \in \mathbb{Z}_+, x_4, x_5, x_6 \in \mathbb{R}_+. \end{aligned} \tag{3.17}$$

We have $\eta = \frac{1}{2}$, $\zeta = -\frac{3}{4}$, $\eta^C = 3$ and $\zeta^C = -1$. Consequently, we have the following lower and upper bounding functions on the value function of (3.17) (see Figure 3.1).

$$F_L(d) = \begin{cases} \frac{1}{2}d & \text{if } d \geq 0 \\ \frac{3}{4} & \text{if } d < 0 \end{cases} \quad \text{and} \quad F_U(d) = \begin{cases} 3d & \text{if } d \geq 0 \\ -d & \text{if } d < 0 \end{cases} \tag{3.18}$$

□

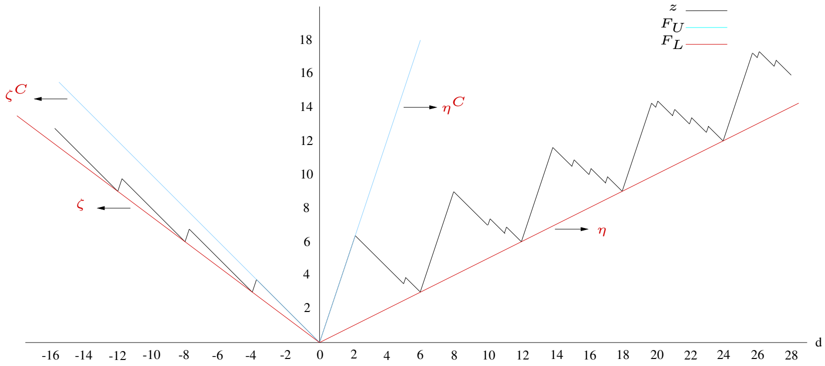


Figure 3.1: Bounding functions on the value function of (3.17).

Not only do the functions F_U and F_L bound the value function, but the following proposition shows that the bounds are always tight.

Proposition 3.7 *Let F_L and F_U be defined as in (3.14) and (3.16) and let*

$$\begin{aligned} d_U^+ &= \sup\{d \geq 0 \mid z(d) = F_U(d)\}, \\ d_U^- &= \inf\{d \leq 0 \mid z(d) = F_U(d)\}, \\ d_L^+ &= \inf\{d > 0 \mid z(d) = F_L(d)\}, \quad \textit{and} \\ d_L^- &= \sup\{d < 0 \mid z(d) = F_L(d)\}. \end{aligned}$$

Then

(i) $z(d) = F_U(d)\ \forall d \in (d_U^-, d_U^+)$,

(ii) $\eta < \eta^C \iff \{d \mid z(d) = F_U(d) = F_L(d), d \in \mathbb{R}_+\} \equiv \{0\} \iff d_U^+ < \infty$,

(iii) $\zeta > \zeta^C \iff \{d \mid z(d) = F_U(d) = F_L(d), d \in \mathbb{R}_-\} \equiv \{0\} \iff d_U^- > -\infty$,

(iv) $d_L^+ \geq d_U^+$ *if* $d_L^+ > 0$, *and similarly,* $d_L^- \leq d_U^-$ *if* $d_L^- < 0$,

(v) if $b \in \{d \in \mathbb{R} \mid z(d) = F_L(d)\}$, *then* $z(kb) = kF_L(b)\ \ \forall k \in \mathbb{Z}_+$.

Proof.

(i) We first consider the interval $[0, d_U^+)$. If $\eta^C = \infty$, then $d_U^+ = 0$ because otherwise there would exist $d > 0$ with $z(d) = F_U(d) = \infty$, which is a contradiction to our initial assumption that z is finite everywhere. Therefore, assume that $\eta^C < \infty$ and that for $b_1 > 0$, we have $F_U(b_1) > z(b_1)$. Note that if we can show that $F_U(d) > z(d)\ \forall d > b_1$, then we must have $z(d) = F_U(d)\ \forall d \in [0, d_U^+)$. Let $x^1 \in \mathbb{Z}_+^I \times \mathbb{R}_+^C$ be an optimal solution to (P) with right-hand side b_1. Then,

$$F_U(b_1) = \eta^C b_1 > z(b_1) = cx^1. \tag{3.19}$$

Let $t \in C$ be such that $\eta^C = \frac{c_t}{a_t}$. Then, setting $\gamma = \sum_{\substack{i=1 \\ i \neq t}}^{n} a_i x_i^1$, we have

$$x_t^1 = \frac{b_1 - \gamma}{a_t}.$$

Next, let $b_2 > b_1$. In this case, the vector $x^2 \in \mathbb{Z}_+^I \times \mathbb{R}_+^C$ defined by

$$x_i^2 = x_i^1\ \forall i \in N, i \neq t \ \text{ and } \ x_t^2 = \frac{b_2 - \gamma}{a_t}$$

is a feasible solution to the MILP with right-hand side b_2. This implies that

$$cx^2 = z(b_1) + c_t \left(\frac{b_2 - \gamma}{a_t} - \frac{b_1 - \gamma}{a_t} \right) = z(b_1) + \eta^C (b_2 - b_1) \geq z(b_2).$$

Therefore, by (3.19), we have

$$F_U(b_2) = \eta^C b_2 = \eta^C b_1 + (\eta^C b_2 - \eta^C b_1) > z(b_1) + \eta^C(b_2 - b_1) \geq z(b_2).$$

Hence, we have the result. Similar arguments can be made for the interval $(d_U^-, 0]$.

(ii) Since F_L and F_U are linear on $\mathbb{R}_+$ and $F_L(d) = \eta d \leq z(d) \leq \eta^C d = F_U(d)\ \forall d \in \mathbb{R}_+$, then it is clear that $\eta < \eta^C \iff \{d \mid z(d) = F_U(d) = F_L(d), d \in \mathbb{R}_+\} \equiv \{0\}$.

For the remaining part, we will show that $\eta < \eta^C \iff d_U^+ < \infty$. Now, if $\eta = \eta^C$, then $z(d) = F_U(d) = F_L(d)\ \forall d \in \mathbb{R}_+$ and hence $d_U^+ = \infty$. On the other hand, if $\eta < \eta^C$, then for $t \in I$ such that $\frac{c_t}{a_t} = \eta$, we have

$$c_t \geq z(a_t) \geq \eta a_t = c_t$$

and thus, $F_L(a_t) = z(a_t)$ and $a_t > 0$. If $d_U^+ = \infty$, then from (i), we should also have $z(a_t) = F_U(a_t)$. However, this is a contradiction to our previous assumption that $\eta < \eta^C$.

(iii) The proof follows from arguments similar to those made above.

(iv) Assume that $d_L^+ > 0$ and $d_L^+ < d_U^+$. From (i) and our discussion in the proof of (ii), we then have $z(d) = F_U(d) = F_L(d)\ \forall d \in \mathbb{R}_+$ and hence, $d_L^+ = 0$, which is a contradiction. The proof of the second part follows from similar arguments.

(v) The proof follows from the fact that

$$kF_L(b) = kz(b) \geq z(kb) \geq F_L(kb) = kF_L(b)\ \forall k \in \mathbb{Z}_+.$$

□

It is also worth mentioning that it is not difficult to show that we must have $F_U(d) = \bar{z}(d)\ \forall d \in \mathbb{R}_+$ if $d_U^+ > 0$ and similarly, $F_U(d) = \bar{z}(d)\ \forall d \in \mathbb{R}_-$ if $d_U^- < 0$, where $\bar{z}$ is the upper d-directional derivative of the value function.

Example 9 Consider the MILP instance (3.17) and its value function. From Proposition 3.7, we have $d_U^+ = 2$, $d_U^- = -1$, $d_L^+ = 6$, $d_L^- = -4$,

$$z(d) = \begin{cases} 2d & \forall d \in [0, 2] \\ -d & \forall d \in [-1, 0] \end{cases},$$

and furthermore,

$$z\,(6k) = 3k \;\text{ and }\; z\,(-4k) = 3k$$

for all $k \in \mathbb{Z}_+$. Observe from Figure 3.2 that these results agree with the value function.

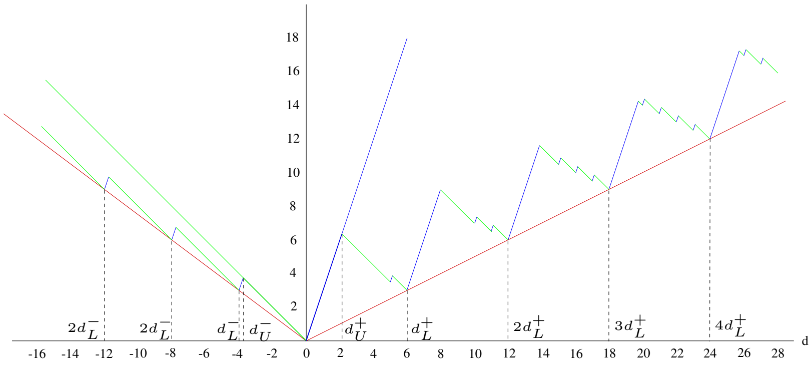


Figure 3.2: Tight relations between the bounding functions and the value function of (3.17).

□

3.2.2 Structure

Recall from Theorem 3.5 that the value function of a general MILP is piecewise-linear and can be represented using the *Jeroslow Formula* as the value function of a related pure integer program and a linear correction term obtained from the coefficients of the continuous variables. Below, we extend these results for the case under consideration and show that z can be represented with at most two continuous variables and consequently, each linear segment of z can be described by a unique integral vector and a slope obtained from the coefficients of these continuous variables. In addition, we analyze the behavior of z at a right-hand side d where it is discontinuous and derive the necessary and sufficient conditions for the continuity of z.

As a first step, let $T \subseteq C$ consist of the indices t^+ and t^- of two continuous variables that achieve the ratios η^C and ζ^C, respectively (if they exist). Formally, let $t^+ \in C$ be such that $\eta^C = \frac{c_{t^+}}{a_{t^+}}$ if $\eta^C < \infty$ and similarly, let $t^- \in C$ be such that $\zeta^C = \frac{c_{t^-}}{a_{t^-}}$ if $\zeta^C > -\infty$. Finally, let $T = \{t^+ \mid \eta^C < \infty\} \cup \{t^- \mid \zeta^C < \infty\}$.

Proposition 3.8 *Let*

$$\begin{aligned} \nu(d) = \min \quad & c_I x_I + c_T x_T \\ s.t. \quad & a_I x_I + a_T x_T = d \\ & x_I \in \mathbb{Z}^I_+, \;\; x_T \in \mathbb{R}^T_+ \end{aligned} \tag{3.20}$$

Then, $\nu(d) = z(d)$ *for all* $d \in \mathbb{R}$.

Proof.

i. $\nu \geq z$: This result follows from the fact that (3.20) is a restriction of (P).

ii. $\nu \leq z$: For a given $b \in \mathbb{R}$, let $\bar{y} \in \mathbb{Z}^I_+$ be the integer part of an optimal solution to MILP (P) with right-hand side b. Then we have

$$\begin{aligned} \nu(b) \;\; & \leq c_I\bar{y} + \;\; \min c_T x \\ & \qquad\qquad a_T x \;=\; b - a_I \bar{y} \\ & \qquad\qquad\quad x \;\in\; \mathbb{R}^T_+ \\ & = c_I\bar{y} + \;\; \max \; u(b - a_I\bar{y}) \\ & \qquad\qquad \zeta^C \leq u \leq \eta^C \\ & = c_I\bar{y} + \;\; u^*(b - a_I\bar{y}) \\ & = z(b) \end{aligned}$$

where

$$u^* = \begin{cases} \eta^C & \text{if} \quad b - a_I\bar{y} > 0, \\ \zeta^C & \text{if} \quad b - a_I\bar{y} < 0, \\ 0 & \text{otherwise} \end{cases}$$

The first inequality follows from the fact that $\bar{y}$ has to be the integer part of a feasible solution to (3.20), because otherwise, the dual problem in the second equation above would be unbounded and hence $\bar{y}$ cannot be an integer part of an optimal solution to MILP (P) with right-hand side d.□

Proposition 3.8 states that we can assume without loss of generality that we have at most two continuous variables, i.e., those indexed by the set T in Proposition 3.20. The rest of the continuous variables can be considered redundant. With this observation, we can simplify the *Jeroslow Formula* to the special case under consideration here.

Let $M \in \mathbb{Z}_+$ be such that $\frac{Ma_j}{a_t} \in \mathbb{Z}$ for all $t \in T$, $j \in I$. Note that such an integer exists by the rationality of a. Define

$$\begin{aligned} g(q) = \min \quad & c_I x_I + \tfrac{1}{M} c_T x_T + z(\varphi) v \\ \text{s.t} \quad & a_I x_I + \tfrac{1}{M} a_T x_T + \varphi v = q \\ & x_I \in \mathbb{Z}_+^I, x_T \in Z_+^T, v \in \mathbb{Z}_+ \end{aligned} \tag{3.21}$$

for all $q \in \mathbb{R}$, where $\varphi = -\frac{1}{M}\sum_{t \in T} a_t$. Furthermore, for $t \in T$, define

$$\omega_t(d) = g(\lfloor d \rfloor_t) + \frac{c_t}{a_t}(d - \lfloor d \rfloor_t) \tag{3.22}$$

for all $d \in \mathbb{R}$, where $\lfloor d \rfloor_t = \frac{a_t}{M}\left\lfloor \frac{Md}{a_t} \right\rfloor$.Then, by Theorem 3.5,

$$z(d) = \min_{t \in T} \omega_t(d) \quad \forall d \in \mathbb{R}. \tag{3.23}$$

Note that both ω_{t^+} and ω_{t^-} are piecewise linear with finitely many linear segments on any closed interval and each of those linear segments has a slope of η^C and ζ^C, respectively. It then follows that z is also piecewise-linear with finitely many linear segments on any closed interval and furthermore, each of those linear segments either coincides with ω_{t^+} and has a slope of η^C or coincides with ω_{t^-} and has a slope of ζ^C. Below, we formally present results that follow from this observation by analyzing the structure of ω_{t^+} and ω_{t^-}.

Theorem 3.9

(i) For $t \in T$, let $b \in \mathbb{R}$ be a breakpoint of ω_t. Then $\frac{Mb}{a_t} \in \mathbb{Z}$.

(ii) ω_{t^+} is continuous from the right and ω_{t^-} is continuous from the left.

(iii) ω_{t^+} and ω_{t^-} are both lower-semicontinuous.

Proof.

(i) Let b be given such that $\frac{Mb}{a_t} \notin \mathbb{Z}$ and let

$$U \equiv \begin{cases} \left[\lfloor b \rfloor_t, \lfloor b \rfloor_t + \frac{a_t}{M}\right), & \text{if } t = t^+ \\ \left(\lfloor b \rfloor_t + \frac{a_t}{M}, \lfloor b \rfloor_t\right], & \text{if } t = t^- \end{cases} .$$

Observe that we have $\lfloor d \rfloor_t = \lfloor b \rfloor_t$ for all $d \in U$ and ω_t is linear over U. Since $b \in U$ and $b \neq \lfloor b \rfloor_t$, we must have that ω_t is continuous at b and therefore b cannot be a breakpoint.

(ii) From the proof of (i), for any $d \in \mathbb{R}$, ω_{t^+} is linear over the interval $\left[\lfloor d \rfloor_t, \lfloor d \rfloor_t + \frac{a_t}{M}\right)$ and therefore continuous from the right. Similarly, ω_{t^-} is linear over the interval $\left(\lfloor d \rfloor_t + \frac{a_t}{M}, \lfloor d \rfloor_t\right]$ and therefore continuous from the left.

(iii) Let $b \in \mathbb{R}$ be a breakpoint of ω_{t^+}. From our discussion in the proof of (i),

$$\omega_{t^+}(d) = \begin{cases} g\left(b - \frac{a_{t^+}}{M}\right) + \frac{c_{t^+}}{a_{t^+}}\left(d - \left(b - \frac{a_{t^+}}{M}\right)\right) & d \in \left[b - \frac{a_{t^+}}{M}, b\right) \quad , \\ g(b) + \frac{c_{t^+}}{a_{t^+}}(d - b) & d \in \left[b, b + \frac{a_{t^+}}{M}\right) \quad . \end{cases}$$

Then we have

$$\begin{aligned} \lim_{\rho \to b^-} \omega_{t^+}(\rho) &= g\left(b - \frac{a_{t^+}}{M}\right) + \frac{c_{t^+}}{a_{t^+}}\left(b - \left(b - \frac{a_{t^+}}{M}\right)\right) \\ &= g\left(b - \frac{a_{t^+}}{M}\right) + \frac{c_{t^+}}{M} \\ &\geq g(b) \\ &= \omega_{t^+}(b) \end{aligned}$$

where the inequality follows the fact that if $(\bar{x}_I, \bar{x}_T, \bar{v})$ is an optimal solution to (3.21) with right-hand side $b - \frac{a_{t+}}{M}$, then $\left(\bar{x}_I, x_{t^+} + 1, \bar{x}_{T\setminus t^+}, \bar{v}\right)$ would be a feasible solution for right-hand side b and the last equality follows from the fact that $b = \lfloor b \rfloor_{t^+}$ by (i). Since from (ii), ω_{t^+} is continuous from the right, we conclude that $\lim_{\rho \to b} \omega_{t^+}(\rho) \geq \omega_{t^+}(b)$ and therefore, ω_{t^+} is lower-semicontinuous.

One can show that ω_{t^-} is also lower-semicontinuous with similar arguments. □

Corollary 3.10

(i) *(Meyer [1975]) z is lower-semicontinuous.*

(ii) *If z is discontinuous at $b \in \mathbb{R}$, then there exists a $\bar{y} \in \mathbb{Z}^I_+$ such that $b - a_I\bar{y} = 0$.*

(iii) *Let U be a maximal interval on which the value function z is linear. If $z(d) = \omega_{t^+}(d)\ \forall d \in U$, then U is closed from the left. On the other hand, if $z(d) = \omega_{t^-}(d)\ \forall d \in U$, then U is closed from the right.*

Proof.

(i) From Theorems 3.5 and 3.9, z is the minimum of two piecewise-linear, lower-semicontinuous functions and therefore, z is also lower-semicontinuous.

(ii) Assume without loss of generality that $b - a_I\bar{y} > 0$. Then we have,

$$\begin{aligned} z(b) &= \lim_{\rho \to b} (c_I\bar{y} + \eta^C(\rho - a_I\bar{y})) \\ &\geq \lim_{\rho \to b} z(\rho) \\ &\geq z(b), \end{aligned}$$

where the first inequality follows from the fact that $\bar{y}$ is the integral part of a feasible solution for any $d \in [b - a_I\bar{y}, \infty)$ and the second inequality from (i) that z is lower-semicontinuous. However, this is a contradiction to the initial statement that z is discontinuous at b since at least one of $\lim_{\rho \to b^+} z(\rho) > z(b)$ and $\lim_{\rho \to b^-} z(\rho) > z(b)$ has to be satisfied.

(iii) Assume that $z(d) = \omega_{t^+}(d)\ \forall d \in U$, U is not closed from the left and let $b = \inf\{d \mid d \in U\}$. Note that we must have $z(b) < \omega_{t^+}(b)$ due to the fact that ω_{t^+} is continuous from the left by Theorem 3.9 and therefore is linear over $\{b\} \cup U$. Since z is discontinuous at b, there exists by (ii) a $\bar{y} \in \mathbb{Z}^I_+$ such that $b - a_I\bar{y} = 0$. Then for $t \in T$ $\frac{Mb}{a_t} \in \mathbb{Z}$, $b = \lfloor b \rfloor_t$ and consequently, we have $z(b) = \omega_t(b)$, which is a contradiction.

Similar arguments can be made to show that U is closed from the right when $z(d) = \omega_{t^-}(d)\ \forall d \in U$.

□

Theorem 3.11 *If the value function z is linear on an interval $U \subset \mathbb{R}$, then there exists a $\bar{y} \in \mathbb{Z}^I_+$ such that $\bar{y}$ is the integral part of an optimal solution for any $d \in U$. Consequently, for some $t \in T$, z can be written as*

$$z(d) = c_I\bar{y} + \frac{c_t}{a_t}(d - a_I\bar{y}) \tag{3.24}$$

for all $d \in U$. Furthermore, for any $d \in U$, we have $d - a_I\bar{y} \geq 0$ if $t = t^+$ and $d - a_I\bar{y} \leq 0$ if $t = t^-$.

Proof. From Theorem 3.5 and our discussion above, we must either have $z(d) = \omega_{t^+}(d)$ or $z(d) = \omega_{t^-}(d)$ for all $d \in U$. Assume the former holds so that from Corollary 3.10, U is closed from the left. Let $(\bar{x}_I, \bar{x}_T)$ be an optimal solution for $b = \min\{d \mid d \in U\}$ with $z(b) = c_I\bar{x}_I + c_T\bar{x}_T$. Since z is linear on U with the slope η^C and passes through the point $(b, z(b))$, we can write $z(d) = g(\lfloor b \rfloor_{t^+}) + \eta^C(d - \lfloor b \rfloor_{t^+})$ for all $d \in U$. For any $q \in U$, we have $q \geq b$ and hence $\left(\bar{x}_I, \bar{x}_{t^+} + \frac{q-b}{a_{t^+}}, \bar{x}_{T\setminus t^+}\right)$ is a feasible solution for right-hand side q. Observe that this is also an optimal solution since

$$\begin{aligned} c_I\bar{x}_I + c_{t^+}\left(\bar{x}_{t^+} + \frac{q-b}{a_{t^+}}\right) + c_{T\setminus t^+}x_{T\setminus t^+} &= c_I\bar{x}_I + c_T\bar{x}_T + c_{t^+}\left(\frac{q-b}{a_{t^+}}\right) \\ &= g(\lfloor b \rfloor_{t^+}) + \eta^C(b - \lfloor b \rfloor_{t^+}) + c_{t^+}\left(\frac{q-b}{a_{t^+}}\right) \\ &= g(\lfloor b \rfloor_{t^+}) + \eta^C(q - \lfloor b \rfloor_{t^+}) \end{aligned}$$

The proof follows similar arguments for the case when z is determined by ω_{t^-}.

For the rest of the claim, note that if $t = t^+$ and $\bar{y}$ is the integral part of an optimal solution for d, then from our discussion in the proof of Theorem 3.20, $d - a\bar{y} \geq 0$ and $z(d) = c_I\bar{y} + \eta^C(d - a_I\bar{y})$. Similarly, if $t = t^-$, then $d - a\bar{y} \leq 0$ and $z(d) = c_I\bar{y} + \zeta^C(d - a_I\bar{y})$. □

Example 10 For the MILP instance (3.17), we have $T = \{4, 5\}$ and hence, x_6 is redundant. Furthermore, for the intervals $U_1 = \left[0, \frac{17}{8}\right], U_2 = \left[\frac{17}{8}, 5\right], U_3 = \left[5, \frac{41}{8}\right], U_4 = \left[\frac{41}{8}, 6\right], \ldots,$ we have $y^1 = (0\ 0\ 0), y^2 = (0\ 1\ 0), y^3 = (0\ 1\ 0), y^4 = (1\ 0\ 0), \ldots$ as the integral parts of the corresponding optimal solutions and therefore, plugging these values to (3.24), we obtain

$$z(d) = \begin{cases} \ldots & \\ 3d & \text{if } d \in U_1 \\ -d + \frac{17}{2} & \text{if } d \in U_2 \\ 3d - \frac{23}{2} & \text{if } d \in U_3 \\ -d + 9 & \text{if } d \in U_4 \\ \ldots & \end{cases}$$

□

As these results show, the continuous variables indexed by t^+ and t^-, and the associated quantities η^C and ζ^C are key components of the structure of the value function. It turns out that we can further state conditions on the continuity of z using these components.

Theorem 3.12

(i) $\eta^C < \infty$ *if and only if* z *is continuous from the right. Similarly,* $\zeta^C > -\infty$ *if and only if* z *is continuous from the left.*

(ii) Let $U \subset \mathbb{R}$*,* $V \subset \mathbb{R}$ *be the intervals that any two consecutive linear segments of* z *are defined on and let* $\alpha_U, \alpha_V \in \{\eta^C, \zeta^C\}$ *be the corresponding slopes. Then* $\alpha_U \neq \alpha_V$ *if and only if both* η^C *and* ζ^C *are finite if and only if* z *is continuous everywhere.*

Proof.

(i) Assume that $\eta^C < \infty$. Now, if $\zeta^C = -\infty$, then $z(d) = \omega_{t^+}(d)\ \forall d \in \mathbb{R}$ and from Theorem 3.9, z is continuous from the right. Otherwise, let $bin\mathbb{R}$ be such that z is discontinuous at b. Then from Corollary 3.10, there exists a $\bar{y} \in \mathbb{Z}^I_+$ such that $b - a_I\bar{y} = 0$. In this case, both $\frac{Mb}{a_{t^+}}$ and $\frac{Mb}{a_{t^-}}$ are integral and therefore, $b = \lfloor b \rfloor_t, t \in T$ and $z(b) = \omega_{t^+}(b) = \omega_{t^-}(b)$. However, this is a contradiction, since ω_{t^+} is continuous from the right and ω_{t^-} is continuous from the left and therefore by lower-semicontinuity, z would then have to be continuous at b.

On the other hand, assume that z is continuous from the right. If $\eta^C = \infty$, then $z(d) = \omega_{t^-}(d)\ \forall d \in \mathbb{R}$, z is continuous from the left and z can only be continuous from the right when ω_{t^-} is linear over $\mathbb{R}$. From Proposition 3.7 that, this is possible only when $\eta^C = \zeta^C$, which is a contradiction.

Similar arguments can be made to show that the second part of the claim is also valid.

(ii) The proof follows directly from (i).

□

Theorem 3.12 states that the continuity of z depends only on the finiteness of η^C and ζ^C and that when z is continuous, the slopes of the linear segments of z alternate between these two values. Consequently, pursuant to the results of Proposition 3.7, it is not hard to show that we must have $\eta^C = \infty \iff d^+_U = 0$ and $\zeta^C = -\infty \iff d^-_U = 0$. In other words, z overlaps with F_U in a right-neighborhood of the origin if and only if $\eta^C < \infty$ and similarly, in a left-neighborhood of the origin if and only if $\zeta^C > -\infty$.

Example 11 Consider the following MILP instance with a fixed right-hand side b

$$\begin{aligned} \min \quad & x_1 - \tfrac{3}{4}x_2 + \tfrac{3}{4}x_3 \\ \text{s.t} \quad & \tfrac{5}{4}x_1 - x_2 + \tfrac{1}{2}x_3 = b \quad \text{and} \\ & x_1, x_2 \in \mathbb{Z}_+, x_3 \in \mathbb{R}_+. \end{aligned} \tag{3.25}$$

As seen in Figure 3.3, the value function of problem (3.3) is continuous only from the right. For the intervals $U_1 = \left[0, \frac{1}{4}\right), U_2 = \left[\frac{1}{4}, \frac{1}{2}\right), U_3 = \left[\frac{1}{2}, \frac{3}{4}\right), U_4 = \left[\frac{3}{4}, 1\right), U_5 = \left[1, \frac{3}{2}\right), U_6 = \left[\frac{3}{2}, \frac{7}{4}\right), U_7 = \left[\frac{7}{4}, 2\right), U_8 = \{2\}$, we have $y^1 = (0\ 0), y^2 = (1\ 1), y^3 = (2\ 2), y^4 = (3\ 3), y^5 =$

$(4\ 4), y^6 = (2\ 1), y^7 = (3\ 2), y^8 = (4\ 3)$ as the integral parts of the corresponding integral solutions. Note that z is discontinuous at $d_1 = 0, d_2 = \frac{1}{4}, d_3 = \frac{1}{2}, d_4 = 1, d_5 = \frac{5}{4}, d_6 = \frac{3}{2}, d_7 = \frac{7}{4}, d_8 = 2$ in the interval $[0, 2]$ and for each discontinuous point, we have $d_i - \left(\frac{5}{4}y_1^i - y_2^i\right) = 0$. Furthermore, observe that $d_U^- = 0$, $d_U^+ = \frac{1}{4}$ and each linear segment has the slope of $\eta^C = \frac{3}{2}$. □

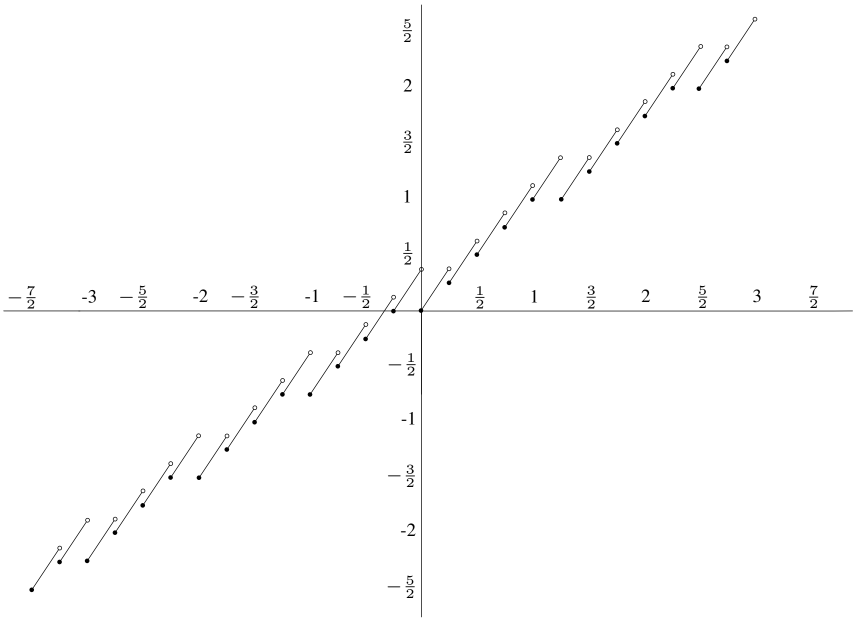


Figure 3.3: The value function of (3.25).

3.2.3 Maximal Subadditive Extension

Let a real-valued function f be subadditive on the interval $[0, h], h > 0$. The *maximal subadditive extension* of f to $\mathbb{R}_+$ is the function f_S defined for each $d \in \mathbb{R}_+$ by

$$f_S(d) = \begin{cases} f(d) & \text{if } d \in [0, h] \\ \inf\limits_{\mathcal{C} \in \mathcal{C}(d)} \sum\limits_{\rho \in \mathcal{C}} f(\rho) & \text{if } d > h\ , \end{cases} \tag{3.26}$$

where $\mathcal{C}(d)$ is the set of all finite collections $\{\rho_1, ..., \rho_R\}$ such that $\rho_i \in [0, h], i = 1, ..., R$ and $\sum_{i=1}^R \rho_i = d$. Each collection $\{\rho_1, ..., \rho_R\}$ is called an *h-partition* of d.

Theorem 3.13 *(Bruckner [1960]) Let $f : [0,h] \rightarrow \mathbb{R}$ be a subadditive function with $f(0) = 0$ and let f_S be defined as in (3.26). Then f_S is subadditive and if g is any other subadditive extension of f to $\mathbb{R}_+$, then $g \leq f_S$.*

Note that because of subadditivity, we can find the exact size of the h-partitions needed to be evaluated to get the value of $f_S(d)$ for any $d > h$.

Theorem 3.14 *For any $d > h$, let $k_d \geq 2$ be the integer such that $d \in \left(\frac{k_d}{2}h, \frac{k_d+1}{2}h\right]$. Then*

$$f_S(d) = \inf\{\sum_{i=1}^{k_d} f(\rho_i) \mid \sum_{i=1}^{k_d} \rho_i = d, \rho_i \in [0,h], i = 1,...,k_d\}. \tag{3.27}$$

Proof. Without loss of generality, consider an h-partition of d, $\{\rho_1, ..., \rho_{k_d+1}\}$, where k_d is defined as above. In this case, at least one pair of members of this partition must have a sum less than or equal to h, since otherwise, $k_d \sum_{i=1}^{k_d+1} \rho_i = k_d d > \frac{(k_d+1)k_d}{2}h$, which would be a contradiction to the fact that $d \leq \frac{k_d+1}{2}h$. Hence, there is another h-partition with the same sum and one less member, which means that any partition with more than k_d members is redundant. □

Using Theorem 3.14, we can formulate a procedure to construct f_S recursively for all $d \in \mathbb{R}_+$ as follows:

i. $f_S(d) = f(d), d \in [0,h]$. Let $p := h$.

ii. For any $d \in \left(p, p + \frac{p}{2}\right]$, let

$$f_S(d) = \inf\{f_S(\rho_1) + f_S(\rho_2) \mid \rho_1 + \rho_2 = d, \rho_1, \rho_2 \in (0,p]\} \tag{RP}$$

Let $p := p + \frac{p}{2}$ and repeat this step.

Observe that this procedure extends the function over an interval that is the half of the previous interval so that we only need to consider p-partitions with size 2 for any $d \in \left(p, p + \frac{p}{2}\right]$ by Theorem 3.14.

One special case occurs when f_S has the property that $f_S(kh + d) = kf(h) + d$ for all $d \in [0, h]$ and all $k \in \mathbb{Z}_+$, i.e., the function "repeats" itself over intervals of size h. In particular, this property is observed whenever $f_S(h + d) = f(h) + f(d)$ for all $d \in (0, h]$ (Laatsch [1964]). The following result states necessary and sufficient conditions for this behavior.

Proposition 3.15 *(Barton and Laatsch [1966]) Let f be a subadditive function on $[0, h]$. Then $f_S(kh + d) = kf(h) + d$ for all $d \in [0, h]$ and all $k \in \mathbb{Z}_+$ if and only if for all $\rho_1 \in (0, h]$,*

$$f(\rho_1) \leq f(h + \rho_1 - \rho_2) - f(h) + f(\rho_2) \tag{3.28}$$

for all ρ_2 satisfying $\rho_1 \leq \rho_2 \leq h$.

Corollary 3.16 *Let f be a subadditive function with $f(0) = 0$ that is concave on $[0, h]$. Then $f_S(kh + d) = kf(h) + f(d)$ for all $d \in (0, h]$ and all $k \in \mathbb{Z}_+$.*

All the results above are also valid when extending a subadditive function f defined on the interval $[h, 0]$ for $h < 0$ to $\mathbb{R}_-$.

We now state one of our main results: if $F(d) = z(d)$ for all d in a certain neighborhood of the origin that we will define, then the maximal subadditive extension of F to all of $\mathbb{R}$ must be precisely the value function. Before we proceed, we first note that we can change the "inf" to "min" in (3.26) if the seed function is the value function itself.

Lemma 3.17 *Let the function $f : [0, h] \rightarrow \mathbb{R}$ be defined by $f(d) = z(d)\ \forall d \in [0, h]$. Then,*

$$f_S(d) = \begin{cases} z(d) & \textit{if } \ d \in [0, h] \\ \min\limits_{\mathcal{C} \in \mathcal{C}(d)} \sum\limits_{\rho \in \mathcal{C}} z(\rho) & \textit{if } \ d > h\ , \end{cases} \tag{3.29}$$

Proof. Assume that this is not the case and we have for $d > h$

$$\inf_{\mathcal{C} \in \mathcal{C}(d)} \sum_{\rho \in \mathcal{C}} z(\rho) < \sum_{\rho \in \mathcal{C}} z(\rho) \quad \forall \mathcal{C} \in \mathcal{C}(d) \tag{3.30}$$

Note that this is possible only if for some $\mathcal{C} \in \mathcal{C}(d)$ and $\rho \in \mathcal{C}$, we have

$$z(\rho) > \lim_{\sigma \to \rho^+} z(\sigma) \quad \text{or} \quad z(\rho) > \lim_{\sigma \to \rho^-} z(\sigma).$$

However, this is a contradiction to lower-semicontinuity of z given in Corollary 3.10. □

Theorem 3.18 *Let $d_r = \max\{a_i \mid i \in N\}$ and $d_l = \min\{a_i \mid i \in N\}$ and let the functions f_r and f_l be the maximal subadditive extensions of z from the intervals $[0, d_r]$ and $[d_l, 0]$ to $\mathbb{R}_+$ and $\mathbb{R}_-$, respectively. If we define the function*

$$F(d) = \begin{cases} f_r(d) & d \in \mathbb{R}_+ \\ f_l(d) & d \in \mathbb{R}_- \end{cases}, \tag{3.31}$$

then $z = F$.

Proof. $z \leq F$: For a given $d > d_r$ let $\{\rho_i\}_{i=1}^R, \rho_i \in [0, d_r]\ \forall i \in \{1, ..., R\}$ be a d_r-partition of d such that $f_r(d) = \sum_{i=1}^R z(\rho_i)$. Note that such a partition exists by Lemma 3.17. In addition, let $\{x^i\}_{i=1}^R$ be the collection of corresponding optimal solutions. Since $\sum_{i=1}^R x^i$ is a feasible solution to MILP (P) with right-hand side d, then $F(d) = f_r(d) = \sum_{i=1}^R f(\rho_i) \geq z(d)$. Hence, $F(d) \geq z(d)\ \forall d \geq 0$. Similarly, one can also show that $F(d) \geq z(d)\ \forall d < d_l$.

$z \geq F$: Using MILP duality, we show that F is dual feasible.

i. *F is subadditive*: It is enough to show that F is subadditive in $\left[\frac{3d_l}{2}, \frac{3d_r}{2}\right]$ due to recursive procedure (RP). Let $d_1 \in \left[\frac{3}{2}d_l, 0\right]$ and $d_2 \in \left[0, \frac{3d_r}{2}\right]$ be given. From the construction of f_l, we know that $f_l(d_1)$ can be determined by a d_l-partition $\{\rho_i^1\}_{i=1}^R$ of d_1 which has a size of at most 2 and similarly, $f_r(d_2)$ can be determined by a d_r-partition $\{\rho_i^2\}_{i=1}^R$ of d_2 also with a size of at most 2. Then

$$\begin{aligned} F(d_1)+F(d_2) &= \sum_{i=1}^{2} f_l(\rho_i^1)+\sum_{i=1}^{2} f_r(\rho_i^2) \\ &= \sum_{i=1}^{2} z(\rho_i^1)+z(\rho_i^2) \\ &\geq \sum_{i=1}^{2} z(\rho_i^1+\rho_i^2) \end{aligned}$$

Note that the last inequality follows the fact that z is subadditive in $[d_l, d_r]$ and pair sums $\rho_i^1+\rho_i^2 \in [d_l, d_r], i=1,2$. If both these pair sums are either in $[d_l, 0]$ or in $[0, d_r]$, then we are done, since in this case, the subadditivity of f_l or f_r can be invoked directly to show that $\sum_{i=1}^{2} z(\rho_i^1+\rho_i^2) \geq F(d_1+d_2)$.

Otherwise, we have at least one pair sum in each of these intervals. Without loss of generality, let $\rho_1^1+\rho_1^2 \in [d_l, 0]$ and $\rho_2^1+\rho_2^2 \in [0, d_r]$. Then we have

$$\sum_{i=1}^{2} z\left(\rho_i^1+\rho_i^2\right) \quad \geq \quad z\left(\sum_{i=1}^{2} \rho_i^1+\rho_i^2\right)=F(d_1+d_2)$$

since $\sum_{i=1}^{2} \rho_i^1+\rho_i^2 \in [d_l.d_r]$ and z is subadditive in this interval.

ii. $F(a_j) \leq c_j \ \forall j \in I$: Since the range $[d_l, d_r]$ is chosen to include all $a_j, j \in I$, then $F(a_j)=z(a_j) \leq c_j \ \forall j \in I$.

iii. $\bar{F}(a_j)=\limsup_{\delta \to 0^+} \frac{F(\delta a_j)}{\delta} \leq c_j \ \forall j \in C$: For any $j \in C$ and sufficiently small values of $\delta > 0$, we have $F(\delta a_j)=z(\delta a_j) \leq \delta c_j$. This implies that, $\bar{F}(a_j) \leq c_j \ \forall j \in C$.

Also note that $F(0)=z(0)=0$ due to our initial assumptions and thus, F is a feasible dual function and from MILP duality, $z \geq F$. □

3.2.4 Construction

In this section, we will show how to evaluate the value function z explicitly over the entire real line using the properties of z we have described before and the maximal subadditive extension result given in Theorem 3.31. We will construct the value function in two steps: first, we will obtain the value function on $[d_l, d_r]$ and then extend it to the entire real line from this original interval. Without loss of generality, we will only consider constructing the value function over $\mathbb{R}_+$ only. Obtaining z over $\mathbb{R}_-$ can be accomplished similarly after multiplying the coefficient vector a with -1.

Constructing the value function on $[0, d_r]$**.** We begin with the basic approach by considering the size of the minimal interval over which z is linear, from Jeroslow Formula (3.22).

Proposition 3.19 *The minimal size of a maximal interval on which z is linear is $\delta = \min_{t \in T}\{\left\|\frac{a_t}{M}\right\|\}$.*

Furthermore, (3.22) states that if b is a break point, then $b = j\delta$ for some $j \in Z_+$. Then, the straightforward way to obtain z on $[0, d_r]$ would be solving (P) with right-hand side set to $j\delta$ for all $j \in Z_+, j\delta \leq d_r$.

For the discontinuous case, a better approach is to consider the fact that for a break point b, we have from Corollary 3.10 that if $x_I^* \in \mathbb{Z}_+^I$ is the integral part of an optimal solution, then $b - a_I x_I^* = 0$. Hence, if this condition is not satisfied for a given b, then we conclude that $z(d)$ stays linear for all $d \in [a_I x_I^*, b]$. Then, we can use an algorithm that starts from d_r and iterates on linear pieces derived as above.

For the continuous case, however, we can evaluate z with a more efficient algorithm with the following observations:

Proposition 3.20 *Let $d_1, d_2 \in [0, d_r], d_1 < d_2$. For $j \in \{1, 2\}$ and $\sigma \in \{\eta^C, \zeta^C\}$, define the linear functions $\omega_j^\sigma : \mathbb{R} \rightarrow \mathbb{R}$ as follows:*

$$\omega_j^\sigma(d) = \sigma(d - d_j) + z(d_j). \tag{3.32}$$

i. If $\omega_1^\sigma(d_2) = z(d_2)$, then $z(d) = \omega_1^\sigma(d)\ \forall d \in [d_1, d_2]$. Similarly, if $\omega_2^\sigma(d_1) = z(d_1)$, then $z(d) = \omega_2^\sigma(d)\ \forall d \in [d_1, d_2]$.

ii. Let $\bar{d} \in [d_1, d_2]$ be such that $\lambda = \omega_1^{\sigma_1}(\bar{d}) = \omega_2^{\sigma_2}(\bar{d})$, $\sigma_1 \neq \sigma^2, \sigma_1, \sigma_2 \in \{\eta^C, \zeta^C\}$. If $z(\bar{d}) = \lambda$, then $z(d) = \omega_1^{\sigma_1}(d)\ \forall d \in [d_1, \bar{d}]$ and $z(d) = \omega_2^{\sigma_2}(d)\ \forall d \in [\bar{d}, d_2]$.

Proof.

i. Consider the first part of the claim which states that $z(d_1)$ and $z(d_2)$ appear to be on a line segment with the slope σ. Assume that $z(d) \neq \omega_1^{\sigma}(d)$ for some $d \in [d_1, d_2]$. Then, since z is piecewise-linear continuous with alternating slopes in $\{\eta^C, \zeta^C\}$, $z(d_2)$ can not be on the same linear segment with $z(d_1)$, which is a contradiction.

ii. The proof follows from applying the first part on the intervals $[d_1, \bar{d}]$ and $[\bar{d}, d_2]$. □

The first part of Proposition 3.20 considers the structure and the subadditivity property of the value function and states that for any two distinct right-hand sides, if the corresponding optimal solution values can be connected by a line with a slope of η^C or ζ^C, then z has to be linear over the closed interval with the end points of these right-hand sides. Further in the second part, this result is carried out to the case where the original interval is divided into separate intervals considering the intersection points of the linear functions with the slope of η^C and ζ^C and passing through the original end points. Therefore, we can recursively divide $[0, d_r]$ into sub-intervals through the intersection points of each closed intervals and by applying Proposition 3.20 until each linear piece is identified.

Example 12 Consider the MILP instance (3.17). First, we check the intersection point $\bar{d}$ of the lines $3d$ and $-1(d-6) + z(6)$, that pass through the origin with the slope $\eta^C = 3$ and through $(6, z(6))$ with the slope $\zeta^C = -1$. Then, we check whether $z(\bar{d})$ agrees with the intersection value. Since, it is not, we divide $[0, 6]$ into $[0, \bar{d}]$, $[\bar{d}, 6]$ and recursively apply the same procedure (see Figure 3.4). □

Constructing the value function on $\mathbb{R}_+$. Theorem 3.18 states that z can be encoded simply by its two slopes and the set of its breakpoints in the interval $[d_l, d_r]$. Now that we can formulate z as in Theorem 3.18, an obvious question is whether we can algorithmically extract the value $z(d)$ for

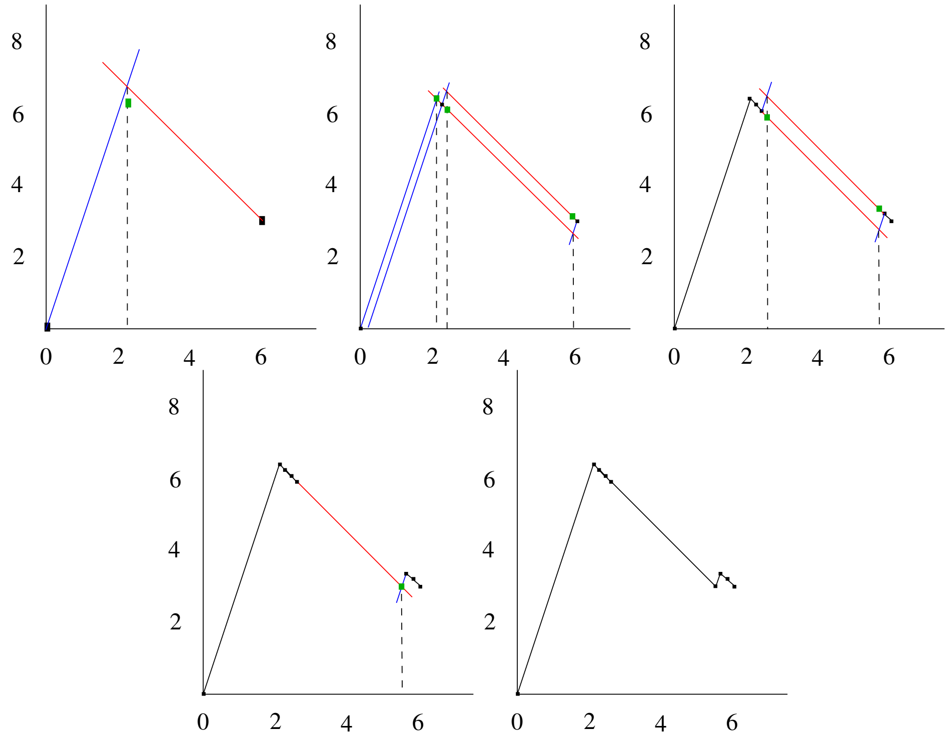


Figure 3.4: Applying Proposition 3.20 to obtain the value function of the MILP instance (3.17) over $[0, 6]$.

a given $d \in \mathbb{R} \backslash [0, d_r]$ from this encoding. The following theorem states that, for a special case, this can be done easily.

Theorem 3.21 *Let d_l, d_r be defined as in Theorem 3.18. If z is concave in $[0, d_r]$, then for any $d \in \mathbb{R}_+$*

$$z(d) = kz(d_r) + z(d - kd_r), \qquad kd_r \leq d < (k+1)d_r \; k \in \mathbb{Z}_+ \tag{3.33}$$

Similarly, if z is concave in $[d_l, 0]$, then for any $d \in \mathbb{R}_-$

$$z(d) = kz(d_l) + z(d - kd_l), \qquad (k+1)d_l < d \leq kd_l \; k \in \mathbb{Z}_+ \tag{3.34}$$

Proof. The proof follows from Corollary 3.16 and Theorem 3.18. □

Example 13 For the MILP instance (3.17), z is concave in $[-4, 0]$ and therefore, for any $d \in \mathbb{R}_-$,

$z(d)$ can be written as in (3.34) which simply requires z to *repeat* itself over the intervals of size -4 (see Figure 3.1). □

In general, however, the question is whether there is a finite algorithm to compute $z(d)$ from the above encoding. Our discussion in the rest of this section shows that indeed there is.

Definitions.

- Let $U_i, i \in \{1, \ldots, s\}$ be the maximal intervals defining the linear segments of z on $[0, d_r]$.
- For each $i \in \{1, \ldots, s-1\}$, let $d^i = \sup\{d \mid d \in U_i\} = \inf\{d \mid d \in U_{i+1}\}$. Setting $d^0 = 0, d^s = d_r$, let $\Phi \equiv \{d^0, \ldots, d^s\}$. We will call the set Φ the *break points* of z on the interval $[0, d_r]$.
- Let $\Psi \equiv \{d^i \mid \alpha^i \leq \alpha^{i+1}, i \in \{1, \ldots, s-1\}\} \cup \{0, d_r\}$ where each α^i is defined as in Theorem 3.12. We will call the set Ψ the *lower break points* of z on the interval $[0, d_r]$. Observe that $\Psi \equiv \Phi$ if z is not continuous everywhere.
- For any $d \in \mathbb{R}_+ \backslash [0, d_r]$, a d_r-partition $\mathcal{C} \in \mathcal{C}(d) \equiv \{\rho_1, \ldots, \rho_{k_d}\}$ is called an optimal partition if $z(d) = \sum_{i=1}^{k_d} z(\rho_i)$ where $k_d \geq 2$ is the integer such that $d \in \left(\frac{k_d}{2} d_r, \frac{k_d+1}{2} d_r\right]$. Note that such a partition exists by Theorem 3.14.

Theorem 3.22 *For any $d \in \mathbb{R}_+ \backslash [0, d_r]$, there is an optimal d_r-partition $\mathcal{C} \in \mathcal{C}(d)$ such that $|\mathcal{C} \backslash \Phi| \leq 1$.*

Proof. Let $\mathcal{C} \equiv \{\rho_1, \ldots, \rho_{k_d}\}$ be an optimal d_r-partition for d. If $|\mathcal{C} \backslash \Phi| \leq 1$, then we are done. Otherwise, there is a pair $\{\rho_l, \rho_r\} \in \mathcal{C} \backslash \Phi$. Without loss of generality, let $\rho_l \leq \rho_r$, $\rho_l \in U_l$ and $\rho_r \in U_r$.

- *z is continuous everywhere*: In this case, both η^C and ζ^C are finite and $\alpha^l, \alpha^r \in \{\eta^C, \zeta^C\}$.
  - $\alpha^l = \alpha^r$: Let $\theta = \rho_l - \min\{d \mid d \in U_l\} = \rho_l - d^{l-1}$ and $\lambda = \max\{d \mid d \in U_r\} - \rho_r = d^r - \rho_r$. Setting $\delta = \min\{\theta, \lambda\}$ and $S \equiv \{\rho_l - \delta, \rho_r + \delta\}$, clearly, $S \cup \mathcal{C} \backslash \{\rho_l, \rho_r\}$ is another optimal d_r-partition for d with $|S \cap \Phi| \geq 1$.

  - $\alpha^l > \alpha^r$: Then, $z(\rho_l - \delta) + z(\rho_r + \delta) < z(\rho_l) + z(\rho_r)$ and therefore, $\mathcal{C}$ cannot be an optimal partition for d.
  - $\alpha^l < \alpha^r$: If we let $\theta = \max\{d \mid d \in U_l\} - \rho_l = d^l - \rho_l$, $\lambda = \rho_r - \min\{d \mid d \in U_r\} = \rho_r - d^{r-1}$, and $\delta = \min\{\theta, \lambda\}$, then $z(\rho_l + \delta) + z(\rho_r - \delta) < z(\rho_l) + z(\rho_r)$ and again, this is a contradiction to the optimality of the partition $\mathcal{C}$.

- *z is not continuous everywhere*: In this case, only one of η^C and ζ^C is finite and either $\alpha^l = \alpha^r = \eta^C < \infty$ or $\alpha^l = \alpha^r = \zeta^C > -\infty$.
  - $\eta^C < \infty$: From Theorem 3.12, we know that each U_i $i \in \{1, \dots, s\}$ is closed from the left and therefore $d^i \in U_{i+1}$ $i \in \{0, \dots, s-1\}$.

    Now, let $\theta = \sup\{d \mid d \in U_l\} - \rho_l = d^l - \rho_l$, $\lambda = \rho_r - \min\{d \mid d \in U_r\} = \rho_r - d^{r-1}$ and $\delta = \min\{\theta, \lambda\}$. If $\theta > \lambda$, then, with $S \equiv \{\rho_l + \delta, \rho_r - \delta\}$, $S \cup \mathcal{C}\backslash\{\rho_l, \rho_r\}$ is another optimal d_r-partition for d with $|S \cap \Phi| \geq 1$. If $\theta \leq \lambda$, then $\mathcal{C}$ can not be an optimal partition for d since $\rho_l + \theta = d^l$ and from lower-semicontinuity of z, $z(\rho_l + \theta) + z(\rho_r - \theta) < z(\rho_l) + z(\rho_r)$.
  - $\zeta^C > -\infty$: With similar arguments, one can show that the claim is still valid for this case.

Note that this procedure updates the optimal partition so that the number of common members of the new optimal partition and Φ increases at least by 1. Therefore, applying the procedure iteratively would yield an optimal partition $\bar{\mathcal{C}} \in \mathcal{C}(d)$, $\bar{\mathcal{C}} \equiv \{\bar{\rho}_1, \dots, \bar{\rho}_{k_d}\}$ so that $|\bar{\mathcal{C}}\backslash\Phi| \leq 1$. □

Example 14 Consider the MILP instance (3.25). We have $d_r = \frac{5}{4}$ and $\Phi = \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, \frac{5}{4}\right\}$. For $b = \frac{17}{8}$, $k_b = 3$ and $\mathcal{C} = \left\{\frac{1}{8}, \frac{3}{4}, \frac{5}{4}\right\}$ is an optimal d_r-partition with $|\mathcal{C}\backslash\Phi| = 1$. □

In fact, we can further restrict the search space for the continuous case by only considering the *lower break points*. Note that this would be expected, considering Corollary 3.16.

Corollary 3.23 *If both η^C and ζ^C are finite, then for any $d \in \mathbb{R}_+\backslash[0, d_r]$ there is an optimal d_r-partition $\mathcal{C} \in \mathcal{C}(d)$ such that $|\mathcal{C}\backslash\Psi| \leq 1$.*

Proof. Assume that $\mathcal{C} \equiv \{\rho_1, \ldots, \rho_{k_d}\}$ is an optimal d_r-partition for d such that ρ_r, $\rho_l \in \mathcal{C}\backslash\Psi$. From Theorem 3.22, we know that at least one of ρ_r, ρ_l is in Φ.

- $\rho_r, \rho_l \in \Phi$: Since z is continuous, then $\rho_r \in U_r \cap U_{r+1}$ with $\alpha^r > \alpha^{r+1}$ and $\rho_l \in U_l \cap U_{l+1}$ with $\alpha^l > \alpha^{l+1}$. Let $\lambda = \rho_r - \min\{d \mid d \in U_r\} = \rho_r - d^{r-1}$, $\theta = \max\{d \mid d \in U_{l+1}\} - \rho_l = d^{l+1}$ and $\delta = \min\{\theta, \lambda\}$. Clearly, $\mathcal{C}$ cannot be an optimal partition since for any ϵ with $0 < \epsilon < \delta$, $\{\rho_l + \epsilon, \rho_r - \epsilon\} \cup \mathcal{C}\backslash\{\rho_l, \rho_r\}$ is a d_r-partition for d with $z(\rho_l + \epsilon) + z(\rho_r - \epsilon) < z(\rho_l) + z(\rho_r)$.
- $\rho_r \in \Phi$ and $\rho_l \notin \Phi$: Similarly, assume that $\rho_r \in U_r \cap U_{r+1}$ with $\alpha^r > \alpha^{r+1}$ and $\rho_l \in U_l$.
  - $\alpha^l = \eta^C$: Let $\theta = \rho_l - \min\{d \mid d \in U_l\} = \rho_l - d^{l-1}$, $\lambda = \max\{d \mid d \in U_{r+1}\} - \rho_r = d^{r+1} - \rho_r$, $\delta = \min\{\theta, \lambda\}$. Then, $\mathcal{C}$ can not be an optimal partition for d since for any ϵ with $0 < \epsilon < \delta$, $\{\rho_l - \epsilon, \rho_r + \epsilon\} \cup \mathcal{C}\backslash\{\rho_l, \rho_r\}$ is a d_r-partition for d with $z(\rho_l - \epsilon) + z(\rho_r + \epsilon) < z(\rho_l) + z(\rho_r)$.
  - $\alpha^l = \zeta^C$: Similarly, let $\theta = \max\{d \mid d \in U_l\} - \rho_l$, $\lambda = \rho_r - \min\{d \mid d \in U_r\}$, $\delta = \min\{\theta, \lambda\}$. Then again, $\mathcal{C}$ can not be an optimal partition for d since for any ϵ with $0 < \epsilon < \delta$, $\{\rho_l + \epsilon, \rho_r - \epsilon\} \cup \mathcal{C}\backslash\{\rho_l, \rho_r\}$ is a d_r partition for d and $z(\rho_l + \epsilon) + z(\rho_r - \epsilon) < z(\rho_l) + z(\rho_r)$.

Therefore, and from Theorem 3.22, we conclude that $|\mathcal{C}\backslash\Psi| \leq 1$. □

Example 15 Consider the MILP instance (3.17). For the interval $[0, 6]$, we have $\Psi = \{0, 5, 6\}$. For $b = \frac{31}{2}$, $\mathcal{C} = \{5, 5, \frac{11}{2}\}$ is an optimal d_r-partition with $|\mathcal{C}\backslash\Psi|$ = 1. □

From Theorem 3.22 and Corollary 3.23, we conclude that we need only to search a finite space of d_r-partitions to obtain the value of $z(d)$ for any $d \in \mathbb{R}_+\backslash[0, d_r]$. In particular, we only need to consider the collection set

$$\Lambda(d) \equiv \{\mathcal{C} \mid \mathcal{C} \in \mathcal{C}(d), |\mathcal{C}| = k_d, |\mathcal{C}\backslash\Psi| \leq 1\}, \tag{3.35}$$

which can be written equivalently as

$$\Lambda(d) \equiv \{\mathcal{H} \cup \{\mu\} \mid \mathcal{H} \in \mathcal{C}(d-\mu), |\mathcal{H}| = k_d - 1, \mathcal{H} \subseteq \Psi, \sum_{\rho \in \mathcal{H}} \rho + \mu = d, \mu \in [0, d_r]\} \quad (3.36)$$

In other words,

$$z(d) = \min_{\mathcal{C} \in \Lambda(d)} \sum_{\rho \in \mathcal{C}} z(\rho) \quad (3.37)$$

Observe that the set $\Lambda(d)$ is finite, since Ψ is finite due to the fact that z has finitely many linear segments on $[0, d_r]$ and since for each $\mathcal{H} \in \Lambda(d)$, μ is uniquely determined. In particular, from our discussion in the proof of Theorem 3.9,

$$\Psi \leq \min_{t \in T} \left\{ M \left| \frac{d_r}{a_t} \right| \right\} \quad (3.38)$$

due to the fact that for any $d \in [0, d_r]$, w_t is linear over the interval $\left[\lfloor d \rfloor_t, \lfloor d \rfloor_t + \frac{a_t}{M}\right)$ if $t = t^+$ and $\left(\lfloor d \rfloor_t + \frac{a_t}{M}, \lfloor d \rfloor_t\right]$ if $t = t^-$ and $d \in \Psi$ has to be a break point of both w_{t^+} and w_{t^-} if η^C and ζ^C are finite. Consequently, since we are looking for d_r partitions of size $k_d - 1$ with members chosen from the set Ψ,

$$\Lambda(d) \leq \begin{pmatrix} k_d + \Psi - 2 \\ \Psi - 1 \end{pmatrix}. \quad (3.39)$$

With these observations, we can reformulate the recursive procedure (RP) in a more useful way. Setting $\Psi([0, p])$ to the set of the lower break points of z in the interval $(0, p]$ $p \in \mathbb{R}_+$, we have

i. Let $p := d_r$.

ii. For any $d \in \left(p, p + \frac{p}{2}\right]$, let

$$z(d) = \min\{z(\rho_1) + z(\rho_2) \mid \rho_1 + \rho_2 = d, \rho_1 \in \Psi([0, p]), \rho_2 \in (0, p]\} \quad \text{(RP2)}$$

Let $p := p + \frac{p}{2}$ and repeat this step.

In fact, we can also write the middle step of (RP2) as follows:

$$z(d) = \min_j g^j(d)\ \forall d \in \left(p, p + \frac{p}{2}\right] \tag{3.40}$$

where, for each $d^j \in \Psi([0,p])$, the functions $g^j : \left[0, p + \frac{p}{2}\right] \to \mathbb{R} \cup \{\infty\}$ are defined as

$$g^j(d) = \begin{cases} z(d) & \text{if } \ d \leq d^j, \\ z(d^j) + z(d - d^j) & \text{if } \ d^j < d \leq p + d^j, \\ \infty & \text{otherwise.} \end{cases} \tag{3.41}$$

Because of subadditivity, we can then write

$$z(d) = \min_j g^j(d)\ \forall d \in \left(0, p + \frac{p}{2}\right]. \tag{3.42}$$

Example 16 Consider the MILP instance (3.17). For the left side of the origin, we have only one lower break point -4 (other than origin) and therefore z repeats itself over intervals of size -4. Note that this result is parallel to the conclusion of Corollary 3.16. For the right side of the origin, we have $\Psi([0,6]) = \{5,6\}$. Then,

$$g^1(d) = \begin{cases} z(d) & 0 \leq d \leq 5, \\ 4 + z(d-5) & 5 < d \leq 9. \end{cases} \qquad g^2(d) = \begin{cases} z(d) & 0 \leq d \leq 6, \\ 3 + z(d-6) & 6 < d \leq 9. \end{cases}$$

and consequently (see Figure 3.5) ,

$$z(d) = \min\{g^1(d), g^2(d)\}\ ,\ d \in [0,9]\,.$$

Now that we have obtained z in $[0,9]$, we can apply the same procedure to extend z to $\left[0, \frac{27}{2}\right]$ (see Figure 3.6). □

However, it is still not clear how (RP2) can be used practically to obtain the value function on the extended interval in each step since one also needs to evaluate the set $\Psi([0,p])$ during each

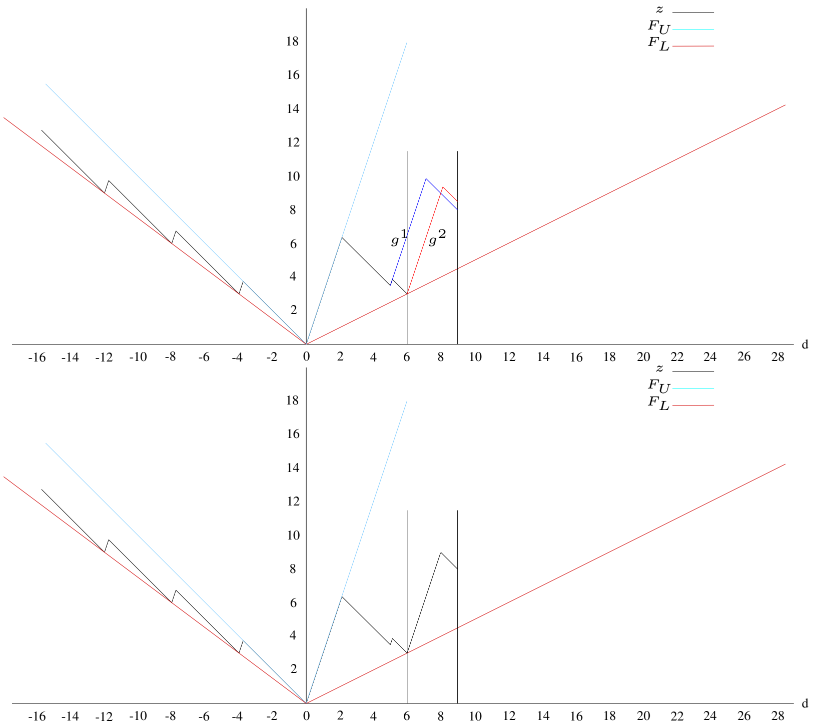


Figure 3.5: Extending the value function of (3.17) from $[0, 6]$ to $[0, 9]$.

iteration. One way to overcome this problem is to iterate the procedure over the set of lower break points, i.e., we can first obtain in the extended interval the lower break points which will naturally also yield the value function on that interval. To achieve this, we first show that each lower-break point in the extended interval can be obtained from the preceding set of lower break points.

Corollary 3.24 *For a given $p \in \mathbb{R}_+$, if d is a lower break point of z on $[p, p + \frac{p}{2}]$, then there exist $\rho_1, \rho_2 \in \Psi([0, p])$ such that $z(d) = z(\rho_1) + z(\rho_2)$ and $d = \rho_1 + \rho_2$.*

Proof. Let $b \in [p, p + \frac{p}{2}]$ be a lower break point of z. Then, $z(b) \leq z(d)$ for all d in a sufficiently small neighborhood of b. Furthermore, let $\rho_1, \rho_2 \in [0, p]$ be such that $z(d) = z(\rho_1) + z(\rho_2)$ by Theorem 3.14 and assume that $\rho_1 \notin \Psi([0, p])$. Clearly, ρ_1 lies on a linear segment and hence, either $z(\rho_1 + \sigma) < z(\rho_1)$ or $z(\rho_1 - \sigma) < z(\rho_1)$ for a sufficiently small $\sigma > 0$. Consequently, we must have $z(\rho_1 + \rho_2 + \sigma) < z(b)$ or $z(\rho_1 + \rho_2 - \sigma) < z(b)$, which violates our initial assumption that b is a lower break point. □

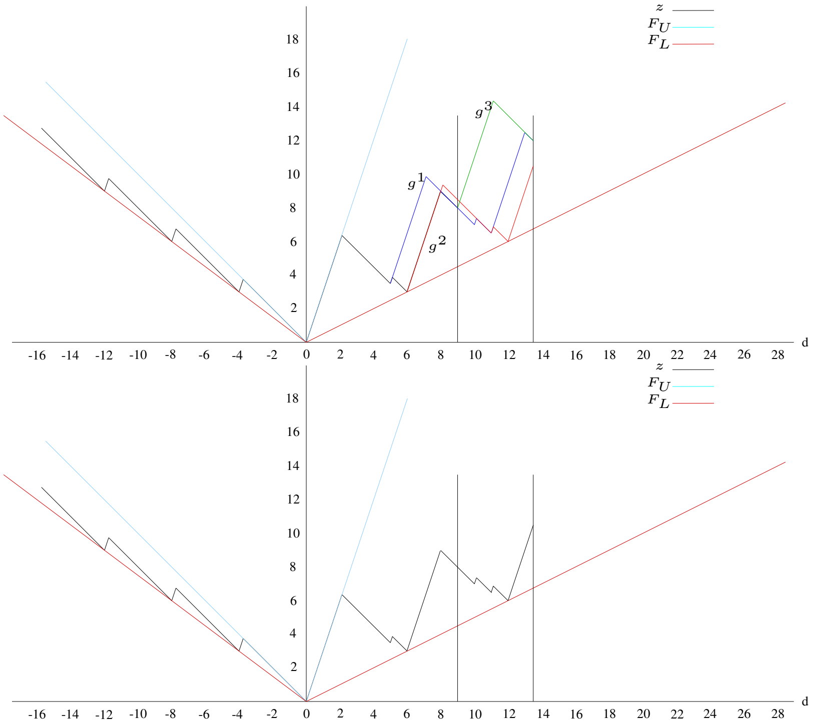


Figure 3.6: Extending the value function of (3.17) from $[0, 9]$ to $\left[0, \frac{27}{2}\right]$.

Considering Corollary 3.24 and setting $\Upsilon([0,p]) \equiv \{z(\rho_1) + z(\rho_2) \mid p < \rho_1 + \rho_2 \leq p + \frac{p}{2}, \rho_1, \rho_2 \in \Psi([0,p])\}$, note that z over the interval $[p, p + \frac{p}{2}]$ is the *lower envelope* function of $\Upsilon([0,p])$ with respect to η^C and ζ^C. This result and the updated recursive procedure are formally given as follows:

Proposition 3.25 *For $p \in \mathbb{R}_+$, let Δ be the collection of functions $f : [p, p + \frac{p}{2}] \rightarrow \mathbb{R}$ such that f is continuous, piecewise-linear with each linear segment having a slope of either η^C or ζ^C and $\Psi_f \subseteq \Upsilon([0,p])$ where Ψ_f is the set of lower break points of f. Furthermore, let $g \in \Delta$ be a minimal function with respect to Δ, that is, for any $d \in [p, p + \frac{p}{2}]$, $g(d) \leq f(d)$ for all $f \in \Delta$. Then, g is unique and $z(d) = g(d) \forall d \in [p, p + \frac{p}{2}]$.*

i. Let $p := d_r$.

ii. Set $\Upsilon([0,p]) \equiv \{z(\rho_1) + z(\rho_2) \mid p < \rho_1 + \rho_2 \leq p + \frac{p}{2}, \rho_1, \rho_2 \in \Psi((0,p])\}$ and let

$$z(d) = g(d) \forall d \in [p, p + \frac{p}{2}] \tag{RP3}$$

where g is the subadditive lower envelope function of $\Upsilon([0,p])$ and is given as in Proposition 3.25. Let $p := p + \frac{p}{2}$ and repeat this step.

Periodicity. Note that the recursive formula (RP3) does not include a termination criteria and, in this sense, is not a finite procedure. An obvious question is whether there is a finite representation of the value function on the real line and how we can make our recursive procedure finite. In this section, we show that the value function indeed always repeats itself in a sense we will define outside of a finite interval and derive the necessary and sufficient conditions to detect this periodicity during the recursive algorithm.

Let $\mathcal{D} = \{d \mid z(d) = F_L(d)\}$. Note that $\mathcal{D} \neq \emptyset$ by Proposition 3.7. Furthermore, let $h_r = \min\{d \in \mathcal{D} \mid d \geq d_r\}$ and observe that $d_r \leq h_r < 2d_r$. Define the functions $f_j : \mathbb{R}_+ \rightarrow \mathbb{R}, j \in \mathbb{Z}_+ \backslash \{0\}$ as follows

$$f_j(d) = \begin{cases} z(d) & , d \leq jh_r \\ kz(h_r) + z(d - kh_r) & , d \in ((k+j-1)h, (k+j)h], k \in \mathbb{Z}_+ \backslash \{0\}. \end{cases} \tag{3.43}$$

Proposition 3.26

i. $f_j(d) \geq f_{j+1}(d) \geq z(d)$ *for all* $d \in \mathbb{R}_+, j \in \mathbb{Z}_+ \backslash \{0\}$.

ii. For $s \in \mathbb{Z}_+ \backslash \{0\}$, $f_s(d) = z(d)\ \forall d \in \mathbb{R}_+$ *if and only if* $f_s(d) = f_{s+1}(d)\ \forall d \in \mathbb{R}_+$.

Proof.

i. For $d \in \mathbb{R}_+, j \in \mathbb{Z}_+ \backslash \{0\}$,

- $d \leq jh_r$: It is clear from (3.43) that $f_j(d) = f_{j+1}(d) = z(d)$.

- $jh_r < d$: Let $k \in \mathbb{Z}_+ \backslash \{0\}$ be such that $d \in ((k+j-1)h_r, (k+j)h_r]$. Then,

  $$f_j(d) = kz(h_r) + z(d - kh_r) \geq kz(h_r) + z(d - kh_r + h_r) - z(h_r) = f_{j+1}(d) \geq z(d) \,,$$

  where the first inequality follows from the subadditivity of the value function z and the last inequality from the additional fact that $(k-1)z(h_r) = z((k-1)h_r)$ since $h_r \in \mathcal{D}$.

ii. Assume that there exists $s \in \mathbb{Z}_+ \backslash \{0\}$ such that $f_s(d) = z(d) \; \forall d \in \mathbb{R}_+$. Then from the first part, it is clear that $f_s(d) = f_j(d) \; \forall d \in \mathbb{R}_+, j > s, j \in \mathbb{Z}_+$.

On the other hand, assume that $f_s(d) = f_{s+1}(d) \; \forall d \in \mathbb{R}_+$ for some $s \in \mathbb{Z}_+ \backslash \{0\}$. Then

$$z(d) = f_s(d) = f_{s+1}(d) = z(h_r) + z(d - h_r) \; \forall d \in [sh_r, (s+1)h_r]. \tag{3.44}$$

Let $\mathcal{C} = \{\rho_1, \ldots, \rho_R\}$ be an optimal d_r-partition for a given $d \in ((s+1)h_r, (s+2)h_r]$. Note that since $h_r \geq d_r$, there exists $\bar{\mathcal{C}} \subset \mathcal{C}$ such that $\bar{d} = \sum_{\rho \in \mathcal{C} \backslash \bar{\mathcal{C}}} \rho \in [sh_r, (s+1)h_r]$ and we have

$$\begin{aligned} z(d) &= \sum_{\rho \in \mathcal{C} \backslash \bar{\mathcal{C}}} z(\rho) + \sum_{\rho \in \bar{\mathcal{C}}} z(\rho) \\ &\geq z(\bar{d}) + \sum_{\rho \in \bar{\mathcal{C}}} z(\rho) \\ &= z(h_r) + z(\bar{d} - h_r) + \sum_{\rho \in \bar{\mathcal{C}}} z(\rho) \\ &\geq z(h_r) + z(d - h_r) \\ &\geq z(d) \end{aligned}$$

by (3.44) and subadditivity of z. It follows then $z(d) = z(h_r) + z(d - h_r) \forall d \in ((s+1)h_r, (s+2)h_r]$ and hence, $f_{s+2}(d) = f_{s+1}(d) = f_s(d) \forall d \in \mathbb{R}_+$. Similarly, one can show that $f_j(d) = f_s(d) \forall d \in \mathbb{R}_+, j > s, j \in \mathbb{Z}_+$ and that from first part, $z(d) = f_s(d) \forall d \in \mathbb{R}_+$. □

Example 17 For the sample problem (3.17), the functions $f_j, j = 1, \ldots, 4$ are given in Figures

3.7 and 3.8. Observe that $h_r = 6, f_4(d) = f_5(d)\ \forall d \in \mathbb{R}_+$ and from Proposition 3.26, $z(d) = f_4(d)\ \forall d \in \mathbb{R}_+$ where

$$f_4(d) = \begin{cases} z(d) & , d \leq 24 \\ kz(6) + z(d-6k) & , d \in (6(k+3), 6(k+4)], k \in \mathbb{Z}_+ \backslash \{0\}. \end{cases} \tag{3.45}$$

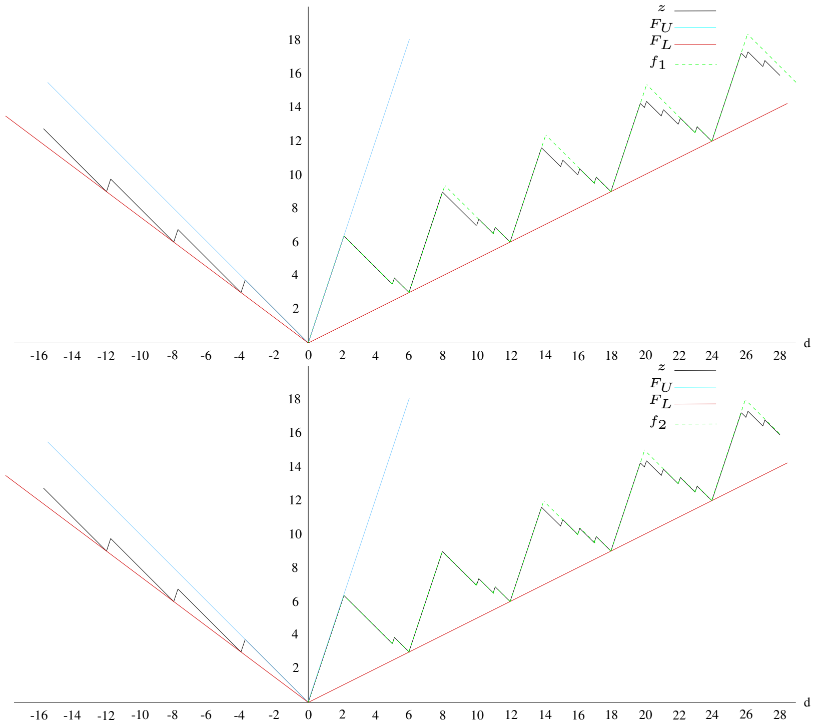


Figure 3.7: Relation between the functions f_1, f_2 and the value function z.

□

Note that Proposition 3.26 only characterizes the periodicity property of the value function, if it exists, over the upper bounding functions f_j and states that the sufficient condition to detect it is to find two consecutive upper bounding functions f_s and f_{s+1} formulated by (3.43) that are equal. Next, we show that such functions do always exist and therefore, the value function *repeats* itself over the intervals of size h_r outside of interval $[0, sh_r]$ for some $s \in \mathbb{Z}_+$.

Lemma 3.27 *For $d \in \mathbb{R}_+$, $\Psi([0,d]) \subset \mathbb{Q}$.*

Proof. The proof follows from the rationality of a and Theorem 3.11. □

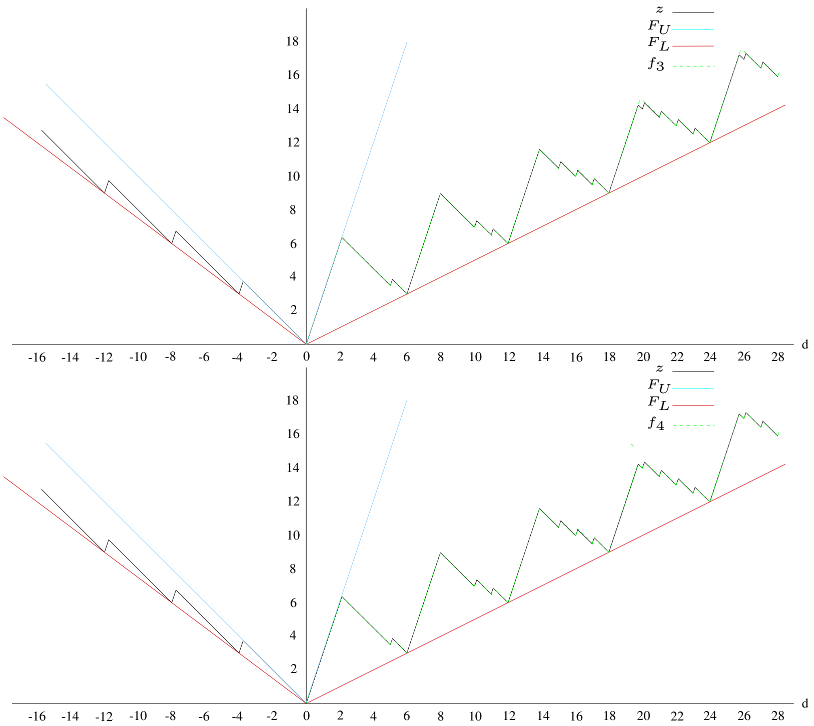


Figure 3.8: Relation between the functions f_3, f_4 and the value function z.

Theorem 3.28 *There exists $s \in \mathbb{Z}_+, s > 0$ such that $z(d) = f_s(d)\ \forall d \in \mathbb{R}_+$.*

Proof. For each $d_i \in \Psi([0, d_r])$, let

$$\mathcal{T}_i = \{k \in \mathbb{Z}_+ \backslash \{0\} \mid kz(d_i) \geq tz(h_r) + z(\rho), th_r + \rho = kd_i, t \in \mathbb{Z}_+ \backslash \{0\}, \rho \in [0, d_r]\}. \quad (3.46)$$

Note that $\mathcal{T}_i \neq \emptyset$ since $d_i, h_r \in \mathbb{Q}_+$ by Lemma 3.27 and therefore, there exists $k_i, t_i \in \mathbb{Z}_+ \backslash \{0\}$ such that $k_i d_i = t_i h_r$ and

$$k_i z(d_i) \geq z(k_i d_i) \geq F_L(k_i d_i) = F_L(t_i h_r) = t_i z(h_r)$$

by the subadditivity of z and Proposition 3.7. Now, set $\lambda_i = \min\{k \mid k \in \mathcal{T}_i\}$ and $s = \min\{k \in \mathbb{Z}_+ \mid (k-1)h_r > \sum_i (\lambda_i - 1)d_i\}$. Next, observe that for a given $d \in [sh_r, (s+1)h_r]$, there is an optimal d_r-partition $\mathcal{C} = \{\rho_1, \ldots, \rho_R\}$ such that for some $d_s \in \Psi([0, d_r])$, $q = \sum_{i=1}^{R} \sigma_i \geq \lambda_s$

where

$$\sigma_i = \begin{cases} 1 & \text{if} \rho_i = d_s \\ 0 & \text{otherwise} \end{cases} \qquad i = 1, \ldots, R\,.$$

Then we have

$$\begin{aligned} z(d) &= qz(d_s) + \sum_{\rho \in \mathcal{C}, \rho \neq d_s} z(\rho) \\ &= \lambda_s z(d_s) + (q - \lambda_s) z(d_s) + \sum_{\rho \in \mathcal{C}, \rho \neq d_s} z(\rho) \\ &\geq \lambda_s z(d_s) + z((q - \lambda_s) d_s) + \sum_{\rho \in \mathcal{C}, \rho \neq d_s} z(\rho) \\ &\geq t_s z(h_r) + z(\rho_s) + z((q - \lambda_s) d_s) + \sum_{\rho \in \mathcal{C}, \rho \neq d_s} z(\rho) \\ &\geq t_s z(h_r) + z(d - t_s h_r) \\ &\geq z(h_r) + z(d - h_r) \\ &\geq z(d) \end{aligned}$$

where $t_s \in \mathbb{Z}_+, \rho_s \in [0, d_r]$ are the parameters that satisfy $\lambda_s z(d_s) \geq t_s z(h_r) + z(\rho_s)$ by (3.46). Therefore, we conclude that $z(d) = z(h_r) + z(d - h_r)\ \forall d \in [sh_r, (s+1)h_r]$, that is, $f_s(d) = f_{s+1}(d)\ \forall d \in \mathbb{R}_+$, and that $z(d) = f_s(d)\ \forall d \in \mathbb{R}_+$ by Proposition 3.26. □

Though it is not straightforward to evaluate, the main characterization of f_s over all $d > sh_r$ is the fact that it can be obtained from the subadditive function, which repeats itself over the intervals of size h_r, is between z and F_L, and is maximal with respect to such functions. First though, we give two results regarding the behavior of such functions.

Proposition 3.29 *(Barton and Laatsch [1966]) Let f be a real-valued subadditive function with $f(0) = 0$ and defined on $[0, h]$. Then $f_S(kh+d) = kf(h)+f(d)$ for all $d \in [0, h]$ and all $k \in \mathbb{Z}_+$ if and only if for all $\rho_1 \in (0, h]$,*

$$f(\rho_1) \leq f(h + \rho_1 - \rho_2) - f(h) + f(\rho_2) \tag{3.47}$$

for all ρ_2 *satisfying* $\rho_1 \leq \rho_2 \leq h$.

Proposition 3.30 *Let* Δ_S *be the collection of all subadditive functions* $f : [0, h_r] \rightarrow \mathbb{R}$ *such that* $F_L(d) \leq f(d) \leq z(d) \forall d \in [0, h_r]$ *and* $f_S(kh_r + d) = kf(h_r) + f(d)$ *for all* $d \in [0, h_r]$ *and all* $k \in \mathbb{Z}_+$. *Furthermore, let* $g \in \Delta_S$ *be a maximal function with respect to* Δ_S, *that is, for any* $d \in [0, h_r]$, $g(d) \geq f(d)$ *for all* $f \in \Delta_S$. *Then,* g *is unique and there exists* $s \in \mathbb{Z}_+$ *such that* $z(d) = g(d) \forall d \geq sh_r$.

Proposition 3.31 *Define* $f[0, h] : \rightarrow \mathbb{R}$ *as follows*

$$f(d) = z(qh + d) - qz(h) \quad d \in [0, h]. \tag{3.48}$$

Then

(i) $z(d) \geq f(d) \geq F_L(d) \; \forall d \in [0, h]$,

(ii) f *is subadditive,*

(iii) the maximal subadditive extension of f *is*

$$f_S(d) = kf(h) + f(d - kh) \forall d > h, d \in (kh, (k+1)h], k \in \mathbb{Z}_+ \backslash \{0\},$$

(iv) $f(d) = z(d) \; \forall d \geq qh$ *where* $q \in \mathbb{Z}_+ \backslash \{0\}$ *such that* $z(d) = g_q(d) \; \forall d \in \mathbb{R}_+$.

Proof.

(i) For any $d \in [0, h]$, we have

$$\begin{aligned} z(d) &\geq z(qh + d) - qz(h) \\ &\geq F_L(qh + d) - qF_L(h) \\ &= qF_L(h) + F_L(d) - qF_L(h) \\ &= F_L(d). \end{aligned}$$

where the first inequality follows from the subadditivity of z.

(ii) Similarly, let $d_1, d_2, d_1 + d_2 \in [0, h]$. Then

$$\begin{aligned} f(d_1) + f(d_2) &= z(qh + d_1) - qz(h) + z(qh + d_2) - qz(h) \\ &\geq z(2qh + d_1 + d_2) - 2qz(h) \\ &= g_q(2qh + d_1 + d_2) - 2qz(h) \\ &= z(qh + d_1 + d_2) - qz(h) \\ &= f(d_1 + d_2). \end{aligned}$$

(iii) For all $d_1 \in (0, h]$ and $d_2 \in [d_1, h]$, we have

$$\begin{aligned} f(d_2) + f(h + d_1 - d_2) &= z(qh + d_2) + z(qh + h + d_1 - d_2) - 2qz(h) \\ &\geq z(2qh + h + d_1) - 2qz(h) \\ &= z(qh + d_1) + z(h) - qz(h) \\ &= f(d_1) + f(h), \end{aligned}$$

and therefore, from Proposition 3.29, $f_S(d) = kf(h) + f(d - kh) \forall d > h, d \in (kh, (k + 1)h], k \in \mathbb{Z}_+0$.

(iv) Note that for all $d \geq qh$, we have

$$f_S(d) = kz(h) + z(qh + d - kh) - qz(h) = (k - q)z(h) + z(d - (k - q)h) = g_q(d)$$

where $d \in (kh, (k + 1)h], k \in \mathbb{Z}_+0$.

□

Proposition 3.31 states that $z(d)$ converges to the subadditive function that is between z and F_L and maximal with respect to (3.47).

Example 18 For the sample problem (3.17), $q = 2$, $h = 3$ and $f(d) = z(d + 6) - 2 \; \forall d \in [0, 3]$.

The maximal extension of f is then

$$f_S(d) = k + f(d - 3k)\ \forall d > 3, d \in (3k, (k+1)3], k \in \mathbb{Z}_+ \backslash \{0\} \tag{3.49}$$

and given in Figure 3.9. Observe that f is the maximal subadditive function that can be obtained below z and above F_L on the interval $[0, h]$ with the property (3.47). □

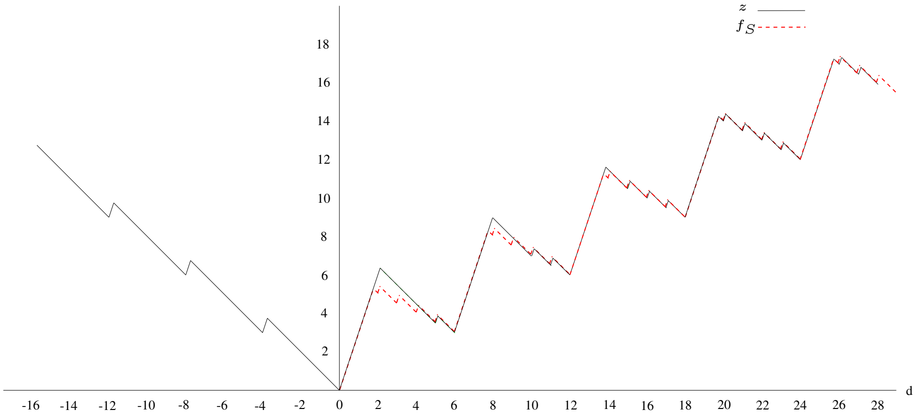


Figure 3.9: Maximal subadditive extension of the function (3.48) for the sample problem (3.17).

With Theorem 3.28, we can make the recursive procedure (RP3) finite by adding a termination step by detecting the periodicity of the value function:

i. Let $p := 2h_r$.

ii. Set $\Upsilon([0,p]) \equiv \{z(\rho_1) + z(\rho_2) \mid p < \rho_1 + \rho_2 \leq p + h_r, \rho_1, \rho_2 \in \Psi((0,p])\}$ and let

$$z(d) = g(d) \forall d \in [p, p + h_r] \tag{RP4}$$

where g is the subadditive lower envelope function of $\Upsilon([0,p])$.

iii. If $z(d) = z(h_r) + z(d - h_r)$ for all $d \in \Psi([p, p + h_r])$, then stop.

iv. Otherwise, let $p := p + h_r$ and repeat the last two steps.

Note in (RP4) that though it is not necessary, we start the procedure from the interval $[0, 2h_r]$ and increase the current interval size at each step by h_r in order to simplify the presentation of detection of the periodicity in the last step. In addition, we check in the termination step whether Theorem 3.28 is satisfied only over the lower break points of $[p, p + h_r]$, since it is apparent that the lower break points together with η^C and ζ^C are sufficient to obtain the value function in this interval.

Finally, we redefine the set $\Upsilon([0, p])$ to include only the pair sums that appear only in the current extended interval $p+h_r$. Observe that Proposition 3.25 is still satisfied since we initially start from the interval $[0, 2h_r]$ to have $p + \frac{p}{2} > h_r$ in each iteration. In fact, we can further restrict the search space and the cardinality of $\Upsilon([0, p])$ using the properties of maximal subadditive extension and the set $\mathcal{D}$.

Proposition 3.32

i. *For a given $k \geq 2, k \in \mathbb{Z}_+$, $z(d) = \min\{z(\rho_1) + z(\rho_2) \mid \rho_1 + \rho_2 = d, \rho_1 \in [0, 2h_r], \rho_2 \in [(k-1)h_r, kh_r]\} \; \forall d \in [kh_r, (k+1)h_r]$.*

ii. *Furthermore, if d is a lower break point of z on $[kh_r, (k+1)h_r]$, then $z(d) = \min\{z(\rho_1) + z(\rho_2) \mid \rho_1 + \rho_2 = d, \rho_1 \in \Psi([0, 2h_r]), \rho_2 \in \Psi([(k-1)h_r), kh_r])\}$.*

Proof.

i. Let $\mathcal{C} = \{\rho_1, \ldots, \rho_R\}$ be an optimal d_r-partition for a given $d \in [kh, (k + 1)h]$. Note that since $h \geq d_r$, there exists $\bar{\mathcal{C}} \subset \mathcal{C}$ such that $\bar{d} = \sum_{\rho \in \mathcal{C} \setminus \bar{\mathcal{C}}} \rho \in [(k-1)h, kh]$, $d - \bar{d} \in [0, 2h]$ and hence, the proof is complete.

ii. The result follows from the first part and Corollary 3.24.

□

Proposition 3.32 states that the lower break points and hence $z(d) \;\; d \in [kh_r, (k + 1)h_r]$ can be obtained by checking the lower break points on the intervals $[0, 2h_r]$ and $[(k - 1)h_r, kh_r]$. Combining this result with Proposition 3.25, we get:

i. Let $p := 2h_r$.

ii. Set $\Upsilon([0,p]) \equiv \{z(\rho_1) + z(\rho_2) \mid p < \rho_1 + \rho_2 \leq p + h_r, \rho_1 \in \Psi((0, 2h_r]), \rho_2 \in \Psi([p - h_r, p])\}$ and let

$$z(d) = g(d) \forall d \in [p, p + h_r] \tag{RP5}$$

where g is the subadditive lower envelope function of $\Upsilon([0,p])$.

iii. If $z(d) = z(h_r) + z(d - h_r)$ for all $d \in \Psi([p, p + h_r])$, then stop.

iv. Otherwise, let $p := p + h_r$ and repeat the last two steps.

Example 19 Figures 3.10 and 3.11 illustrate the procedure (RP5) for extending the value function of sample problem (3.17) over the intervals $[12, 18]$ and $[18, 24]$.

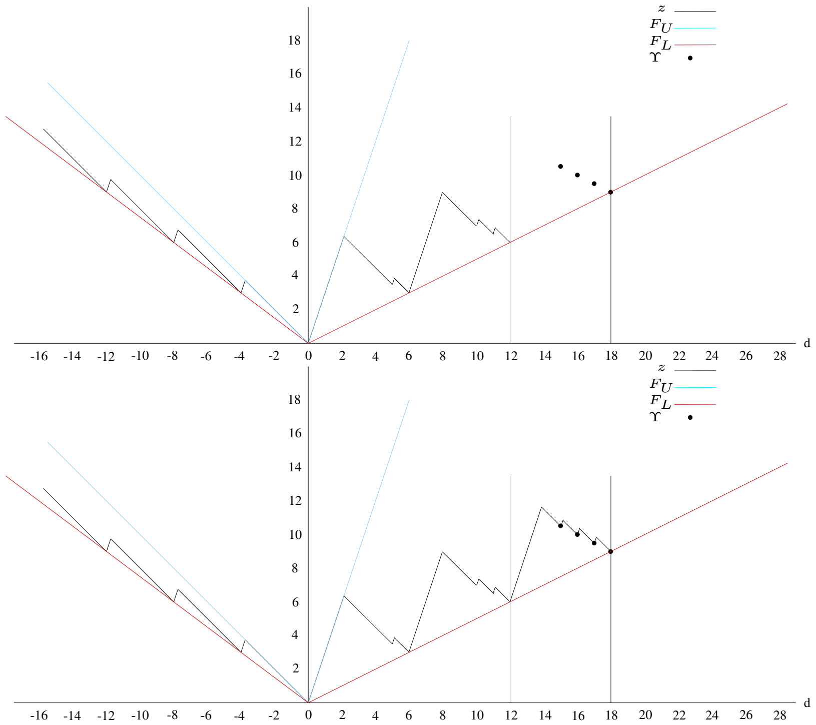


Figure 3.10: Extending the value function of (3.17) from $[0, 12]$ to $[0, 18]$.

□

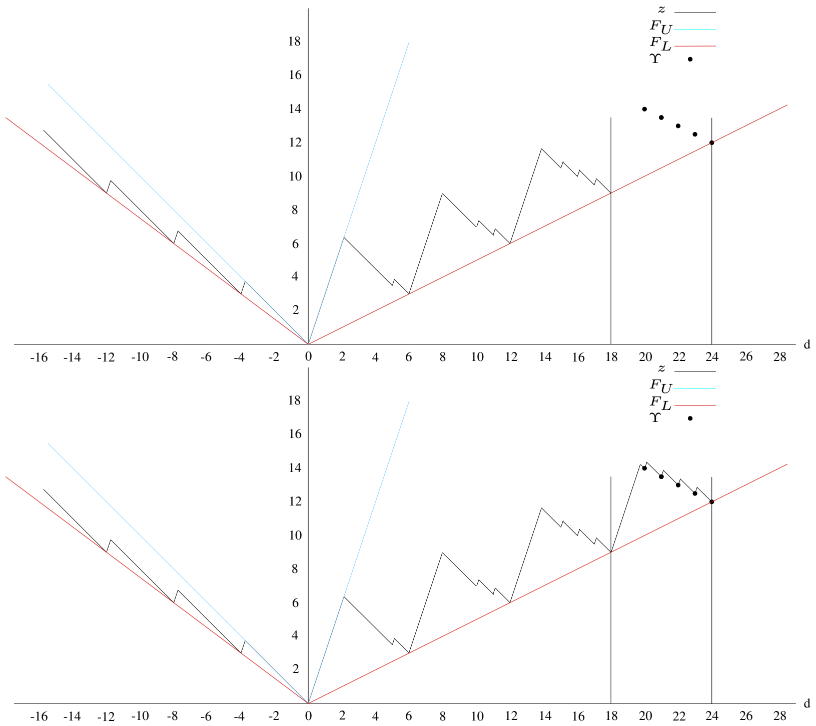


Figure 3.11: Extending the value function of (3.17) from $[0, 18]$ to $[0, 24]$.

3.3 The Value Function of a General MILP

In this section, we get back to the general case. From now on, we use the same notation as in the previous section. However, the notation must be extended to higher dimensional space, unless we state otherwise. That is, we now refer to primal instance (1.7) with $m \geq 1$.

First, we show that some of our results for the single-constraint case can be extended to general case. Considering the Jeroslow Formula (3.4) once again and setting for each $E \in \mathscr{E}$,

$$\omega_E(d) = g(\lfloor d \rfloor_E) + v_E(d - \lfloor d \rfloor_E) \; \forall d \in \mathbb{R}^m \text{ with } \mathcal{S}(d) \neq \emptyset,$$

then we can write

$$z(d) = \min_{E \in \mathscr{E}} \omega_E(d) \; \forall d \in \mathbb{R}^m \text{ with } \mathcal{S}(d) \neq \emptyset.$$

Recall that z is lower-semicontinuous, piecewise-linear with finitely many pieces on any closed interval and each of those segments are defined by the set $\mathscr{E}$. Similarly, one can derive continuity results as in Theorem 3.12. However, it is clear from the fact that $d \in \mathbb{R}^m$ and from the cardinality of $\mathscr{E}$ that combinatorial explosion occurs if one attempts to extend many of the techniques. Therefore, rather than trying to extend all of the results, we only show that maximal extension result is still satisfied, and then we discuss the approximation methods that would be computationally feasible.

3.3.1 Maximal Subadditive Extension

We assume that $A \in \mathbb{Q}^m_+$ and that $S(d) \equiv \mathbb{R}_+$. The first assumption is to ensure the feasible region lies in the nonnegative orthant for simplicity only and the second one is to ensure that the requirements for maximal subadditive extension for (3.26) are satisfied. We emphasize that this assumption is again not crucial in theory due to Theorem 2.5.

Let a real-valued function f be subadditive on the interval $[0, h], h > 0, h \in \mathbb{R}^m$ and the set $M \equiv \{1, \ldots, m\}$. We denote the *maximal subadditive extension* of f as follows:

$$f_S(d) = \begin{cases} f(d) & \text{if } \; d_i \in [0, h_i], i \in M \\ \inf\limits_{\mathcal{C} \in \mathcal{C}(d)} \sum\limits_{\rho \in \mathcal{C}} f(\rho) & \text{if } \; d_i \notin [0, h_i] \text{ for some } i \in M \;, \end{cases} \tag{3.50}$$

where $\mathcal{C}(d)$ is the set of all finite collections $\{\rho^1, ..., \rho^R\}$ such that $\rho^j \in \mathbb{R}^m$, $\rho^j_i \in [0, h_i], j = 1, ..., R$, $i \in M$ and $\sum_{j=1}^{R} \rho^j = d$.

Theorem 3.33 *Let $f : [0, h] \rightarrow \mathbb{R}, h \in \mathbb{R}^m_+$ be a subadditive function with $f(0) = 0$ and let f_S be defined as in (3.50). Then f_S is subadditive and if g is any other subadditive extension of f to $\mathbb{R}_+$, then $g \leq f_S$.*

Next we extend Theorem 3.18 to the general case in order to see that, once again, all the information to obtain the value function is contained in a neighborhood defined by the constraint matrix coefficients.

Theorem 3.34 *Let $q \in \mathbb{Q}^m_+$ be the vector of the maximum of the coefficients of rows of A, that is,*

$$q_i = \max\{a_{ij} \mid j \in N\},$$

and let

$$F(d) = \begin{cases} z(d) & \text{if } \; d_i \in [0, q_i], i \in M \\ \min\limits_{\mathcal{C} \in \mathcal{C}(d)} \sum\limits_{\rho \in \mathcal{C}} z(\rho) & \text{if } \; d_i \notin [0, q_i] \text{ for some } i \in M \;, \end{cases} \tag{3.51}$$

Then, $z(d) = F(d) \;\; \forall d \in \mathbb{R}^m_+$.

Proof. Again, we first note that if the seed function is the value function itself, then "inf" in (3.50) can be changed to "min" since z is lower-semicontinuous. The proof of this result would be same as the proof of 3.17. Therefore, for a given $d \in \mathbb{R}_+$ that z is extended to, there is a q-partition of d such that $F(d) = \sum_{i=j}^R z(\rho^j)$.

$z \leq F$: For a given $d \in \mathbb{R}^m_+$, $d_i \notin [0, q_i]$ for some $i \in M$, let $\{\rho^j\}_{j=1}^R$, $\rho^j \in \times_{i \in M}[0, q_i] \; \forall j \in \{1, ..., R\}$ be a q-partition of d such that $F(d) = \sum_{j=1}^R z(\rho^j)$. In addition, let $\{x^j\}_{j=1}^R$ be the collection of corresponding optimal solutions. Since $\sum_{j=1}^R x^j$ is a feasible solution to MILP (1.7) with right-hand side d, then $F(d) = \sum_{j=1}^R z(\rho^j) \geq z(d)$.

$z \geq F$: Similar to the second part of the proof of Theorem 3.18, F is dual feasible since we choose q such that for all $j \in N$, $A_j \in \times_{i \in M}[0, q_i]$ and hence F is defined by the value function itself for $A_j, j \in N$. □

Under sufficient assumptions, we can further define the concept of a lower break point of the value function. Note that $b \in \mathbb{R}^m_+$ is a break point of z if and only if b is a local minimum of the value function, that is, $z(b) \leq z(b + \theta d)$ for all directions $d \in \mathbb{R}^m_+$ and sufficiently small scalar θ and we can extend for this case the other results that iterate on the break points.

Compared to the single-constraint case, it would undoubtedly prove to be extremely difficult to construct the value function explicitly for the general case due to number of affine pieces needed to describe it. Therefore, we are interested here in approximations that are more easily obtained computationally and could be used in larger applications instead of the value function itself. Note that this is the preferred method even in the LP case for algorithms like Benders decomposition. In the remaining parts, we discuss both lower and upper bounding approximations for the value function and how they can be substituted for the value function in related algorithms.

3.3.2 Lower Bounding Approximations

By definition, a dual function F needs to satisfy $F(d) \leq z(d) \quad \forall d \in \mathbb{R}^m$ and hence, is a lower bounding approximation of the value function. Therefore, when we talk about lower bounding approximations, we are really talking about dual functions. As we have discussed in Section 2.2, although some, like the ones obtained through generating functions and linear representations, are of theoretical interest, it is possible to obtain such functions as a by product of primal solution algorithms like cutting-plane, branch-and-bound, and branch-and-cut methods and our measure of goodness was merely agreement of the dual function with the value function at a single right-hand side. Here, we show that dual functions, though they will not be strong in general with respect to a given right-hand side, can also be constructed from the value functions of the single-constraint relaxations of a general MILP. As we will discuss further, these include not only the single-constraint relaxations, but also aggregated rows and valid inequalities.

We begin with the simplest approach by considering the single-constraint relaxations. Let z_i be the value function of i^{th} row relaxation of (1.7),

$$z_i(p) = \min\{cx \mid a_i x = q, x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+\} \; p \in \mathbb{R}, i \in M \ ,$$

where a_i is the i^{th} row of A. Then clearly, the maximum of these value functions yield a subadditive function approximate to the value function z. Formally, we summarize this result by the following theorem.

Theorem 3.35 *Let $F(d) = \max_{i \in M}\{z_i(d_i)\}, \forall d \in \mathbb{R}^m_+$. Then F is subadditive and $z(d) \geq F(d) \, \forall d \in \mathbb{R}^m_+$.*

Proof. For a pair $d^1, d^2 \in \mathbb{R}^m_+$, assume that $F(d^1 + d^2)$ is attained from the value function of row

j, that is, $F(d^1 + d^2) = z_j(d^1_j + d^2_j)$. Then,

$$F(d^1) + F(d^2) \geq z_j(d^1_j) + z_j(d^2_j) \geq z_j(d^1_j + d^2_j) = F(d^1 + d^2)$$

and hence, F is subadditive. Furthermore, $z(d) \geq z(d_i) \quad \forall d \in \mathbb{R}^m_+, i \in M$ since each value function i is obtained from the corresponding single-constraint relaxation of the original problem. $\square$

Although Theorem 3.35 outlines the basic and easiest way to approximate z through single-constraint relaxations, the resulting function is also the most basic approximation we can obtain. One way to strengthen this result is to apply the maximal subadditive extension algorithm to the subadditive functions obtained by taking the maximum of a subset of the value functions of single-constraint relaxations.

Let $K \subseteq M$ be given, and define $\phi_K : [0, q_K] {\rightarrow} \mathbb{R}$ as

$$\phi_K(h) = \max_{i \in K}\{z_i(h_i)\} \, \forall h \in [0, q_K].$$

Note that ϕ_K is the $|K|$-dimensional subadditive function obtained by the value functions $\{z_i\}_{i \in K}$ and defined over the interval $[0, q_K]$. Now let Θ_K be the maximal subadditive extension of ϕ_K to $\mathbb{R}^{|K|}_+$, that is,

$$\Theta_K(h) = \begin{cases} \phi_K(h) & \text{if} \quad h_i \in [0, q_i], i \in K \\ \inf\limits_{\mathcal{C} \in \mathcal{C}(h)} \sum\limits_{\rho \in \mathcal{C}} \phi_K(\rho) & \text{if} \quad h_i \notin [0, q_i] \text{ for some } i \in K \end{cases}, \tag{3.52}$$

for all $h \in \mathbb{R}^{|K|}_+$ where $\mathcal{C}(h)$ is the set of all finite collections $\{\rho^1, ..., \rho^R\}, \rho^j \in \mathbb{R}^{|K|}_+$ such that $\rho^j \in \times_{i \in K}[0, q_i], j = 1, ..., R$ and $\sum_{j=1}^R \rho^j = h$. Observe that Θ_K is stronger than the function $\max_{i \in S}\{z_i\}$ over $\mathbb{R}^{|K|}_+$ due to the maximality of Θ_K. Combining Θ_K with the rest of the value functions z_i, $i \in M \backslash K$, we can obtain a stronger approximation than the one in Theorem 3.35 that is also subadditive and m-dimensional.

Theorem 3.36 *For a given $K \subseteq M$, let* $F_K(d) = \max\left\{\Theta_K(d_k), \max\limits_{i \in M \backslash K}\{z_i(d_i)\}\right\} \; \forall d \in \mathbb{R}^m_+$, *where Θ_K is defined as (3.52). Then, F_K is subadditive and $z(d) \geq F_K(d) \geq \max_{i \in M}\{z_i(d_i)\}\ \forall d \in \mathbb{R}^m_+$.*

Proof. It is clear from our discussion for the proof of Theorem 3.35 that F is subadditive with $F \geq \max_{i \in M}\{z_i\}$ since it is obtained from the maximal subadditive extension Θ_K of the subadditive function ϕ_K and maximum of the remaining value functions in the set $M \backslash K$. On the other hand, note that Θ_K is a lower bounding function of z on $[0, q_K]$ and from maximal extension result, $z \geq \Theta_K(d_K)\ \ \forall d \in \mathbb{R}^m_+$. □

Another way to obtain single-dimensional relaxations from which approximations to the value function can be readily obtained is through aggregation. Our discussion follows the well-known fact that if some (or all) of the constraints of a primal instance are aggregated into a single constraint then the resulting problem yields a relaxation of the original problem. In particular, let $\omega \in \mathbb{R}^{|K|}$ and set

$$\Theta_K(h, \omega) = \min\{cx \mid \omega A_K x = \omega h, x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+\}\ \forall h \in \mathbb{R}^{|K|}_+\ , \tag{3.53}$$

where A_K is the sub-matrix of A that contains the rows in set K. Then,

$$\Theta_K(h, \omega) \leq \min\{cx \mid A_K x = h, x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+\}\ \forall h \in \mathbb{R}^{|K|}_+. \tag{3.54}$$

Once again, combining Θ_K with the remaining value functions $\{z_i\}_{i \in M \backslash K}$, we can obtain another approximation of z that is subadditive and m-dimensional.

Theorem 3.37 *For a given $K \subseteq M$ and $\omega \in \mathbb{R}^{|K|}$, let*

$$F_K(\omega, d) = \max\left\{\Theta_K(d_K, \omega), \max_{i \in M \backslash K}\{z_i(d_i)\}\right\},\ d \in \mathbb{R}^m_+\ ,$$

where Θ_K is defined as (3.53). Then, F_K is subadditive and $z(d) \geq F_K(\omega, d)\ \forall d \in \mathbb{R}^m_+$.

Proof. The proof follows from Theorem 3.35 and the aggregation result (3.54). □

Finally, we mention that the dual functions generated through the above procedures are general dual functions. In other words, they might or might not be strong for a given right-hand side b, since, unlike the other dual functions, they are not generated from a primal solution procedure for a fixed instance. However, they can still be combined with other dual functions to yield a better approximation of the whole value function. Let the set $\mathscr{F}_1$ be the dual functions obtained from single-constraint relaxations and let the set $\mathscr{F}_2$ be the dual functions obtained from primal solution procedures for each $b \in U \subset \mathbb{R}^m$ where U is some collection of right-hand sides. Clearly,

$$F(d) = \max_{f \in \mathscr{F}_1 \cup \mathscr{F}_2} \{f(d)\} \quad \forall d \in \mathbb{R}^m \tag{3.55}$$

would be a global approximation of the value function. Then, this global approximation can be used to substitute the value function in larger algorithms. Note that the sets $\mathscr{F}_1$ and U can be updated iteratively to get better approximations. In the next section, we give a procedure for approximation that can be used to solve two-stage SIP problems.

3.3.3 Upper Bounding Approximations

Just like the lower bounding approximations, rather than determining an upper bound for a given fixed instance, we are interested in a function that would give a valid upper bound for the value function itself. Since upper bounds are closely related to feasibility, it is harder to obtain such a function than to obtain dual/lower bounding functions.

One possible way is to consider the maximal subadditive extension result for the value function. The idea is that, since the value function is obtained from a neighborhood, then an upper bounding function can also be obtained by some extension rule provided that it bounds the value function from above in the same neighborhood. Before we continue to state this result formally, let us note for a given $g : U \rightarrow \mathbb{R},\ \ U \subset \mathbb{R}^k$ that we call any function G an *extension* of g from U to $\mathbb{R}^k$ if

- $G(d) = g(d) \ \ \forall d \in U$ and,
- for each $d \in \mathbb{R}^k \backslash U$, there exists a collection $\mathcal{C}(d) = \{\rho_1, \ldots, \rho_R\}$ defined in U with the

property $\sum_{\rho\in\mathcal{C}(d)} \rho = d$ and $G(d) = \sum_{\rho\in\mathcal{C}(d)} g(\rho)$.

Theorem 3.38 *For the MILP primal instance (1.7), let q be defined as in Theorem 3.34 and $g : [0, q]\rightarrow\mathbb{R}$ be a real function such that $g(d) \geq z(d) \quad \forall d \in [0, q]$. Then, if G is an extension of g from $[0, q]$ to $\mathbb{R}^m_+$, then $G(d) \geq z(d) \quad \forall d \in \mathbb{R}^m_+$.*

Proof. For a given $d \in \mathbb{R}^m_+\backslash[0, q]$, let $\mathcal{C}(d)$ be the collection that yields $G(d)$. Then,

$$G(d) = \sum_{\rho\in\mathcal{C}(d)} g(\rho) \geq \sum_{\rho\in\mathcal{C}(d)} z(\rho) \geq z(d)$$

by definition of G and the subadditivity of z. □

Note that the upper bounding function G derived by the theorem above is not unique. The quality of G at a right-hand side d solely depends on the extension rule, i.e., choice of the collection $\mathcal{C}(d)$. In this sense, different extension rules yield different upper bounding functions. Obviously, the best of those functions derived by Theorem 3.38 would be obtained when the function to be extended is the value function itself and the rule of extension is maximal subadditive extension. However, although simpler and easier extension rules can be used, it is an open question how to obtain a close upper bounding function in the seed neighborhood.

One other way is to consider *restrictions* of the primal problem. Despite the fact that there are many possible techniques for applying restrictions, we only consider fixing variables in a way that the resulting function, just as in our discussion of dual functions, is a *strong* upper bounding function with respect to a given right-hand side b, i.e., it would give the same bound with the value function at b.

Theorem 3.39 *Let $K \subset N$, $s_i \in \mathbb{R}_+ \;\; i \in K$ be given and define the function $G : \mathbb{R}^m\rightarrow\mathbb{R} \cup \{\infty\}$ such that*

$$G(d) = \sum_{i\in K} c_i s_i + z_{N\backslash K}(d - \sum_{i\in K} A_i s_i) \;\; \forall d \in R^m, \tag{3.56}$$

where A_i is the i^{th} column of A and

$$\begin{aligned} z_{N\backslash K}(h) = \min \quad & \sum_{i \in N\backslash K} c_i x_i \\ s.t \quad & \sum_{i \in N\backslash K} A_i x_i = h \\ & x_i \in \mathbb{Z}_+ \; i \in I, \quad x_i \in \mathbb{R}_+ \; i \in C. \end{aligned} \tag{3.57}$$

Then, $G(d) \geq z(d) \quad \forall d \in \mathbb{R}^m$, if $s_i \in \mathbb{Z}_+ \;\; i \in I \cap K$ and $s_i \in \mathbb{R}_+ \;\; i \in C \cap K$.

Proof. The proof follows the following set of relations

$$\begin{aligned} z(d) &= \min\{c_K x_K + c_{N\backslash K} x_{N\backslash K} \mid A_K x_K + A_{N\backslash K} x_{N\backslash K} = d, \; x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+\} \\ &\leq \min\{c_{N\backslash K} x_{N\backslash K} \mid A_{N\backslash K} x_{N\backslash K} = d - \textstyle\sum_{i \in K} A_i s_i, \; x_i \in \mathbb{Z}_+ \; i \in I, \; x_i \in \mathbb{R}_+ \; i \in C\} \\ &\quad + \textstyle\sum_{i \in K} c_i s_i \\ &= G(d), \end{aligned}$$

as long as $s_i \in \mathbb{Z}_+ \;\; i \in I \cap K$ and $s_i \in \mathbb{R}_+ \;\; i \in C \cap K$. In other words, for $d \in \mathbb{R}^m$, $s_i \; i \in K$ together with an optimal solution $x^*_{N\backslash K}$ (if exists) to (3.57) with right-hand side $d - \sum_{i \in K} A_i s_i$ would constitute a feasible solution to the primal problem (1.7) with right-hand side d. □

Note that one immediate result of Theorem 3.39 is that such an upper bounding function G is strong at a given right-hand side b, if and only if $s_i = x^*_i \quad i \in K$ where x^* is an optimal solution to the primal problem (1.7) with right-hand side b.

Among these type of restrictions, perhaps the most practical one is considering fixing the set of all integer variables, i.e., setting $K = I$. To simplify the presentation and to be able to work on $\mathbb{R}^m$, assume that the dual feasible polyhedron of continuous restriction

$$D_C = \{u \in \mathbb{R}^m \mid uA_C \leq c_C\} \tag{3.58}$$

is non-empty and bounded and let $z_C(d) = \max\{vd \mid v \in V\} \; \forall d \in \mathbb{R}^m$ be the value function of the continuous relaxation of (1.7) where V is the set of extreme points of D_C. We summarize the resulting upper bounding function with the following corollary that follows directly from Theorem 3.39.

Corollary 3.40 *Let x^* be an optimal solution to the primal problem (1.7) with right-hand side set to b and define the function G as*

$$G(d) = c_I x_I^* + z_C(d - A_I x_I^*) \quad \forall d \in \mathbb{R}^m. \tag{3.59}$$

Then, $G(d) \geq z(d) \;\; \forall d \in \mathbb{R}^m$ with $G(b) = z(b)$, and hence, is a strong upper bounding function with respect to b.

The upper bounding function G obtained by (3.59) is closely related to the upper directional derivative of the value function $\bar{z}$ (2.7). Recall from our previous discussion that upper directional derivative is obtained from the continuous variables and is a special output of the corollary above if we choose to set $s_i = 0 \;\; \forall i \in I$. In fact, G is just a translation of $\bar{z}$ and has the the same geometric structure.

Corollary 3.41 *Assume that $z_C(0) = 0$. For a given $b \in \mathbb{R}^m$, $G(d) = z(b) + z_C(d-b) \;\; \forall d \in \mathbb{R}^m$ is a strong upper bounding function with respect to b.*

Proof. The proof follows from the fact that $z_C(0) = 0$ and for $d \in \mathbb{R}^m$,

$$G(d) = z(b) + z_C(d - b) \geq z(b) + z(d - b) \geq z(d).$$

□

Note that each upper bounding function obtained through above methods is not likely to be very close to the value function outside of a local neighborhood of b. However, as in (3.55), we can strengthen the approximation by evaluating strong upper bounding functions for a set of right-hand sides. Let $\mathcal{G}$ be the set of functions obtained by Theorem 3.39 for each $b \in U \subset \mathbb{R}^m$ where U is some collection of right-hand sides. Then

$$G(d) = \min_{g \in \mathcal{G}} \{g(d)\} \quad \forall d \in \mathbb{R}^m \tag{3.60}$$

would be a global upper approximation of the value function.

Example 20 For the MILP problem (3.17), Figure 3.12 shows the strong upper bounding functions $g_i,\ i = 1, \ldots, 5$ obtained from Corollaries 3.40 (g_2, g_4) and 3.41 (g_1, g_3, g_5). In this example, the global upper bounding approximation is $\min\{g_1, \ldots, g5\}$.

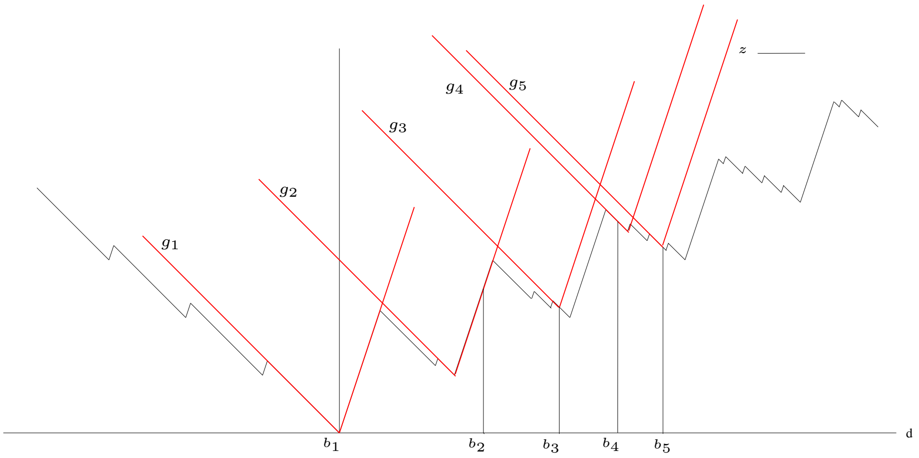


Figure 3.12: Upper bounding functions obtained for (3.17) at right-hand sides $b_i, i = 1, \ldots, 5$.

□

3.4 Applications

In this section we illustrate potential applications for our results on the structure of the value function and its approximations: *two-stage stochastic integer programs* (SIPs) and *mixed integer bilevel programs* (MIBLPs). The common property of these type of optimization problems is the need for the value function of a second stage/lower level MILP problem. An overarching theme of this section is the fact that the value function is not required to be evaluated explicitly for these applications that require implicit knowledge of its structure.

3.4.1 Stochastic Mixed Integer Linear Programming

We consider the following two-stage, stochastic mixed integer program

$$\min\{cx + \mathbf{E}_\xi Q_\xi(x) \mid Ax \geq b, x \in \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1 - r_1}\}, \tag{3.61}$$

where $c \in \mathbb{R}^{n_1}, A \in \mathbb{Q}^{m_1 \times n_1}$ and

$$Q_\xi(x) = \min\{q(\xi)y \mid Wy \geq \omega(\xi) - T(\xi)x, y \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2}\} \tag{3.62}$$

for all ξ, where $W \in \mathbb{R}^{m_2 \times n_2}$. The vector ξ is a random variable from a probability space $(\Xi, \mathscr{F}, \mathcal{P})$ and for each $\xi \in \Xi$, the vectors $q(\xi), \omega(\xi)$ and the rational *technology matrix* $T(\xi)$ have appropriate dimensions. We call $(q(\xi), \omega(\xi), T(\xi))$, which is the realization of the corresponding random variable, a *scenario*.

We assume that the distribution of ξ is discrete and there are $k < \infty$ scenarios. For each scenario (q^j, ω^j, T^j), $j = 1, \ldots, k$, if we define

$$S^j := \{(x, y^j) : Ax \geq b, x \in \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1 - r_1}, T^j x + Wy^j \geq \omega^j, y^j \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2 - r_2}\}, \tag{3.63}$$

then (3.61) can be written equivalently as

$$\min\{cx + \sum_j p^j q^j y^j : (x, y^j) \in S^j, \;\; j = 1, ..., k\} \;, \tag{3.64}$$

where p^j is the probability of scenario j. Since the cardinality of the scenario set might be large enough to cause the number of variables and constraints of (3.64) to be extremely large, conventional methods, such as branch-and-cut, are unlikely to be effective. A mathematical program such as (3.64) would generally be solved by decomposition methods based on introducing the copies of first stage variables through *nonanticipativity constraints* (see Section 4.5.5 for further discussion).

Recently, Kong et al. [2006] presented a value function approach to solve (3.61) with the following additional assumptions:

i. each scenario only consists of the realization of the right-hand side, that is, we fix the second stage objective function and the technology matrix: $q(\xi) = q$, $T(\xi) = T$ for all $\xi \in \Xi$;

ii. the problems in both stages are pure integer, that is, $n_1 = r_1$, $n_2 = r_2$ and all matrices in

(3.61) are integral,

iii. the set $\mathcal{S} = \{x \in \mathbb{Z}^{n_1} \mid Ax \geq b\}$ is nonempty and bounded,

iv. $Q_\xi(x)$ is finite for all ξ and $x \in \mathcal{S}$.

Let us further define the value functions for the problems in both stages as follows:

$$\psi(\beta) = \min\{cx \mid x \in \mathcal{S}_1(\beta)\},\ \mathcal{S}_1(\beta) = \{x\ |Ax \geq b,\ Tx \geq \beta, x \in \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1-r_1}\} \tag{3.65}$$

and

$$\phi(\beta) = \min\{qy \mid y \in \mathcal{S}_2(\beta)\},\ \mathcal{S}_2(\beta) = \{y \mid Wy \geq \beta, y \in Y\} \tag{3.66}$$

for all $\beta \in \mathbb{R}^{m_2}$. Then, we can reformulate (3.61) as

$$\min\{\psi(\beta) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta) \mid \beta \in \mathbf{B}\}\ , \tag{3.67}$$

where $\mathbf{B}$ is the set of *tender variables* and defined as $\mathbf{B} \equiv \{\beta \mid \beta = Tx, x \in \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1-r_1}\}$. Consequently, the relation between (3.61) and (3.67) are given with the following result.

Theorem 3.42 *(Kong et al. [2006]) Let β^* be an optimal solution to (3.67). Then $x^* \in argmin\{cx \mid x \in \mathcal{S}_1(\beta^*)\}$ is an optimal solution to (3.61). Furthermore, the optimal values of the two problems are equal.*

In order to obtain the functions ϕ and ψ, they present two distinct algorithms that are based on the further assumption that all matrices are integral. The first algorithm is an integer programming-based algorithm that solves the corresponding integer programs to obtain $\phi(\beta)$ and $\psi(\beta)$ for certain values of β. For the remaining values of β, they maintain lower and upper bounding functions and iteratively close the gap between these by using the output of the solution procedures for the integer programs and some sensitivity analysis. The second algorithm is based on dynamic programming and employs complete enumeration of feasible β values. With representations of ϕ and ψ computed, they propose two algorithms to solve (3.67). One is a branch-and-bound approach that recursively divides $\mathbf{B}$ into distinct subsets, using arbitrary elements from these subsets, keeps and updates a lower bound on (3.67), and iterates until it is guaranteed that no further partition may increase the current lower bound. The second one is a level-set approach and requires T to be nonnegative. In this approach, they reduce the size of **B** to the set of *minimal tenders* and evaluate (3.67) at each β in this reduced set to find the minimum. They further show that **B** in (3.67) can be restricted to the set of minimal tenders where β is a minimal tender if $\beta = 0$ or $\psi(\beta + e_i) > \psi(\beta)$ for all $i = 1, \ldots, m_2$, where e_i is the i^{th} unit vector. Note that the validity and the termination of the algorithms depend on the finiteness of the set **B** and that **B** $\subset \mathbb{Z}^{m_2}$. Because of this requirement, although Theorem (3.42) still remains valid the algorithms proposed above are not applicable for mixed integer case.

Below, we discuss further improvements to this approach. In the first part, we consider the value function of an MILP with a single constraint to illustrate the general principles. We show how the structure of the value function can be exploited in order to develop an algorithm for SIPs, if it is known explicitly. In the second part, we consider the general case and discuss the ways of substituting the true value function with its lower bounding approximations.

Single-Constraint Case. In this case, we assume that $m_2 = 1$ and that ϕ is continuous everywhere. Therefore, the first stage problem is not constrained to be pure integer, and the set **B** is no constrained to be finite. Note that neither phases of the algorithm presented by Kong et al. [2006] can be applied in this case since, as we mentioned above, evaluating ϕ and ψ in the first phase depends on the necessary assumption that they involve only integer variables, while the branching scheme in the second phase depends on the requirement that **B** $\subset \mathbb{Z}^{m_2}$.

We, likewise, propose a two-phase algorithm. In the first phase, we evaluate ϕ as we have described in the previous section. Observe that ψ is not needed. Instead, we form in the second phase a piecewise-linear optimization problem that can be reformulated as a finite set of MILP subproblems involving only the first stage problem and some auxilary constraints and variables. The algorithm, and therefore the size of each subproblem, is insensitive to the number of scenarios. However, it is sensitive to the structure of the value function ϕ and the number of its linear pieces.

First note that (3.67) is a standard mathematical program with a piecewise linear cost function. Our method of solution is based on standard techniques for modeling such problems. Consider the problem

$$v(f,U) := \min\{cx + f(\beta) \mid Ax \geq b, Tx = \beta, \beta \in U, x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+\} , \tag{3.68}$$

where $f : U \rightarrow \mathbb{R}$, is a continuous piecewise linear function with domain $U \subseteq \mathbb{R}$. A well-known approach to solving this problem is to reformulate (3.68) as an MILP by introducing new variables representing each linear segment of f in terms of its breakpoints.

Case I. Suppose that $U \equiv [\beta_l, \beta_r]$ and f is arbitrary. Let $\beta_l = u_1 < u_2 < \ldots u_t = \beta_r$ be the breakpoints of f on U. Then, any $\beta \in [\beta_l, \beta_r]$ can be uniquely written as

$$\beta = \sum_{i=1}^{t} \lambda_i u_i \ , \tag{3.69}$$

where $\lambda_i \in \mathbb{R}_+$, $i = 1, \ldots, t$, $\sum_{i=1}^{t} \lambda_i = 1$ and $\lambda_i + \lambda_{i+1} = 1$, $i = 1, \ldots, t-1$. In order to formulate (3.68) as an MILP, we can then introduce binary variables y_i, $i = 1, \ldots, t-1$ such that $u_i \leq \beta \leq u_{i+1}$ if and only if $y_i = 1$. We, then get the following equivalent formulation

$$\begin{aligned}
\text{minimize} \quad & cx + \sum_{i=1}^{t} \lambda_i f(u_i) \\
\text{s.t.} \quad & Ax \geq b \\
& Tx = \sum_{i=1}^{t} \lambda_i u_i \\
& \sum_{i=1}^{t} \lambda_i = 1 \\
& \lambda_1 \leq y_1, \ \lambda_t \leq y_{t-1} \\
& \lambda_i \leq y_{i-1} + y_i \quad i = 2, \ldots, t-1 \\
& \sum_{i=1}^{t-1} y_i = 1 \\
& \lambda_i \geq 0, \ x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+, \ y_i \in \{0,1\} \ ,
\end{aligned} \tag{3.70}$$

where the additional constraints ensure that if $y_k = 1$, then $\lambda_i = 0, i \notin \{k, k+1\}$.

Notice that the size of the reformulation depends on the number of the linear segments of f. In some cases, the number of break points might be so large that it would be intractable to solve (3.70) directly. One possible way to overcome this issue is to divide the search space U into subspaces so that the corresponding subproblems would be relatively easy to solve. This result is summarized with the following theorem.

Theorem 3.43 *Let $\{U_j\}$ $j \in \{1, \ldots, R\}$ be a finite collection of closed intervals such that $\cup U_j \equiv U$ and define $f_j : U_j \rightarrow \mathbb{R}$ with $f_j(\beta) = f(\beta), \beta \in U_j$. Then,*

$$\min_j \min\{cx + f_j(\beta) \mid Ax \geq b, Tx = \beta, \beta \in U_j, x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+\} \tag{3.71}$$

has the same optimal solution value as (3.68).

An alternative approach is to consider each convex piece separately and reformulate each subproblem as a piecewise-linear convex optimization problem. Let $\beta_l = \bar{u}_1 < \bar{u}_2 < \ldots \bar{u}_s = \beta_r$ be the breakpoints of f on U such that f is convex on each interval $U_i \equiv [\bar{u}_i, \bar{u}_{i+1}],\ i = 1, \ldots, s-1$. Furthermore, for each interval U_i, let s_i be the number of linear segments of f over U_i and for each $j = 1, \ldots, s_i,\ g_{ij}(\beta) = p_{ij}\beta + \alpha_{ij}$ be the affine linear function representing the linear segment j, where p_{ij} and α_{ij} can be easily obtained from the end points of the corresponding linear segment. Furthermore, for each $i \in \{1, \ldots, s_i\}$, define

$$\begin{aligned} \varphi_i = \text{minimize} \quad & cx + y \\ \text{s.t.} \quad & Ax \geq b \\ & Tx = \beta \\ & y \geq p_{ij}\beta + \alpha_{ij} \qquad j = 1, \ldots, s_i \\ & \bar{u}_i \leq \beta \leq \bar{u}_{i+1} \\ & x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \ . \end{aligned} \tag{3.72}$$

Then, we conclude that

$$\min_i \{\varphi_i\} \tag{3.73}$$

gives the optimal solution value of (3.68).

Case II. Suppose that $U \equiv [\beta_l, \infty)$ and that for a given $\beta_r \in U, \beta_r > \beta_l$

$$f(\beta) = \begin{cases} f(\beta) & ,\beta_l \le \beta \le \beta_r \\ k\Delta + f(\beta - k\delta) & ,\beta \in (k\delta + \beta_l, (k+1)\delta + \beta_l], k \in \mathbb{Z}_+ \backslash \{0\} \end{cases}, \tag{3.74}$$

where $\Delta = f(\beta_r) - f(\beta_l)$ and $\delta = \beta_r - \beta_l$. In other words, f is arbitrary over the interval $[\beta_l, \beta_r]$ and then, repeats itself over the intervals of size δ.

Let again $\beta_l = u_1 < u_2 < \ldots u_t = \beta_r$ be the breakpoints of f on $[\beta_l, \beta_r]$. In this case, any $\beta \in [\beta_l, \infty)$ can be written as

$$\beta = \sum_{i=1}^{t} \lambda_i u_i + y_0 \delta \ , \tag{3.75}$$

where $\lambda_i \in \mathbb{R}_+, i = 1, \ldots, t, \sum_{i=1}^t \lambda_i = 1, \lambda_i + \lambda_{i+1} = 1, i = 1, \ldots, t-1$ and $y_0 \in \mathbb{Z}_+$. Notice also that $f(\beta) = \sum_{i=1}^t \lambda_i f(u_i) + y_0 \Delta$ and therefore, with arguments similar to those above, we obtain the following equivalent MILP

$$\begin{aligned} \text{minimize} \quad & cx + \sum_{i=1}^{t} \lambda_i f(u_i) + y_0 \Delta \\ \text{s.t.} \quad & Ax \ge b \\ & Tx = \sum_{i=1}^{t} \lambda_i u_i + y_0 \delta \\ & \sum_{i=1}^{t} \lambda_i = 1 \\ & \lambda_1 \le y_1, \lambda_t \le y_{t-1} \\ & \lambda_i \le y_{i-1} + y_i \quad i = 2, \ldots, t-1 \\ & \sum_{i=1}^{t-1} y_i = 1 \\ & \lambda_i \ge 0, \ x \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r}, \ y_0 \in \mathbb{Z}_+, \ y_i \in \{0,1\}. \end{aligned} \tag{3.76}$$

As in Case I, if we have an excessive number of break points in $[\beta_l, \beta_r]$, then we can divide (3.76) into relatively manageable subproblems by (3.71).

Now, let us get back to the value function formulation (3.61), which can be written as

$$\min\{cx + \tau(\beta) \mid Ax \geq b, Tx = \beta, x \in \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1 - r_1}\}\ , \tag{3.77}$$

where

$$\tau(\beta) = \sum_j p^j \phi(\omega^j - \beta)\ \beta \in \mathbb{R}. \tag{3.78}$$

Proposition 3.44

i. τ is piecewise-linear continuous function over $\mathbb{R}$.

ii. There exist scalars $\delta_r > 0, \delta_l > 0, \beta_r, \beta_l \in \mathbb{R}$, $\beta_l < \beta_r$ such that τ repeats itself for all $\beta > \beta_r + \delta_r$ and $\beta < \beta_l - \delta_l$ over intervals of size δ_r and δ_l, respectively.

Proof.

i. The proof follows from the fact that ϕ is itself piecewise-linear and continuous over $\mathbb{R}$.

ii. Let $\alpha_r > 0, \alpha_l < 0, \sigma_r, \sigma_l > 0$ be the scalars such that the value function $\phi(\alpha)$ repeats itself over intervals of size σ_r for all $\alpha > \alpha_r + \sigma_r$ and over intervals of size σ_l for all $\alpha < \alpha_l - \sigma_l$, respectively, by Theorem 3.28. Note that for a given scenario j, $\phi(\omega^j - \beta)$ repeats itself for all $\beta > \omega^j - \alpha_l$ over intervals of size α_l and similarly, for all $\beta < \omega^j - \alpha_r$ over intervals of size α_r. Then, setting $\beta_r = \max_j\{\omega^j\} - \alpha_l$, $\beta_l = \min_j\{\omega^j\} - \alpha_r$, $\delta_r = \sigma_l$ and $\delta_l = \sigma_r$ completes the proof.

□

Proposition 3.44 shows that (3.77) can be written using a single piecewise-linear cost function, allowing us to use piecewise-linear optimization as outlined before. To see this, let $U_1 = (-\infty, \beta_l], U_2 = [\beta_l, \beta_r]$ and $U_3 = [\beta_r, \infty)$. Then clearly, the optimal solution value of (3.77) is equal to $\min\{v(\tau, U_1), v(\tau, U_2), v(\tau, U_3)\}$ and $v(\tau, U_2)$ can be obtained from (3.70) whereas $v(\tau, U_1)$ and $v(\tau, U_3)$ from (3.76).

Note that in both formulations, we only require the break points of τ in $[\beta_l - \delta_l, \beta_r + \delta_r]$, which can be enumerated directly from the break points of ϕ and the relative distribution of the scenario values.

Example 21 Consider the SIP instance

$$v = \min\{-3x_1 - 4x_2 + \mathbf{E}_\xi Q_\xi(x) \mid x_1 + x_2 \leq 5,\ x_1 \leq 5,\ x_2 \leq 5, x_1, x_2 \in \mathbb{Z}_+\}\ , \tag{3.79}$$

where

$$\begin{aligned} Q_\xi(x) = \min \quad & 3y_1 + \tfrac{7}{2}y_2 + 3y_3 + 6y_4 + 7y_5 \\ \text{s.t.} \quad & 6y_1 + 5y_2 - 4y_3 + 2y_4 - 7y_5 = \omega(\xi) - 2x_1 - \tfrac{1}{2}x_2 \\ & y_1, y_2, y_3 \in \mathbb{Z}_+, y_4, y_5 \in \mathbb{R}_+. \end{aligned} \tag{3.80}$$

with $\omega(\xi) \in \{6, 12\}$ and probabilities $p_{\omega(\xi)=6} = p_{\omega(\xi)=12} = 0.5$.

The value function of the second stage problem ϕ has been analyzed before and is given in Figure 3.1. Furthermore, the resulting cost function τ given by (3.78) can be written as

$$\tau(\beta) = 0.5(\phi(6 - \beta) + \phi(12 - \beta))\ \ \forall \beta \in \mathbb{R}\ .$$

Note from Proposition 3.44 that τ repeats itself for all $\beta > \beta_r + \delta_r$ and $\beta < \beta_l - \delta_l$ over intervals of size δ_r and δ_l, respectively, where $\beta_r = 16, \delta_r = 4, \beta_l = -18$ and $\delta_l = 6$. The problem, then, reduces to the following optimization problem with a piecewise-linear cost function

$$\begin{aligned} v = \min \qquad & -3x_1 - 4x_2 + \tau(\beta) \\ \text{s.t.} \qquad & x_1 + x_2 \leq 5 \\ & 2x_1 + \tfrac{1}{2}x_2 = \beta \\ & x_1 \leq 5, x_2 \leq 5, x_1, x_2 \in \mathbb{Z}_+\ . \end{aligned}$$

Solving this problem with the techniques described above, we obtain the optimal solution value $v = -15$ with $x_1^* = 2, x_2^* = 3$ and $\beta^* = 5.5$. □

General Case. As we discussed above, the algorithm presented by Kong et al. [2006] requires the fist and second stage value functions ϕ and ψ to be evaluated explicitly. Considering the sizes of the relative domains $\mathcal{S}_1$ and $\mathcal{S}_2$, this approach might be computationally expensive. Below, we outline an alternate dual-based approach that systematically updates approximations of both value functions until the approximation gets close enough to the value function to yield the optimal solution value under the single assumption that the set **B** is finite. In other words, we allow continuous variables in second stage ($r_2 > 0$).

For a given $\hat{\beta} \in \mathbb{R}^{m_2}$, let F_1 be an optimal dual function for the problem

$$\min\{cx \mid Ax \geq b, Tx \geq \hat{\beta}, x \in \mathbb{Z}_+^{n_1}\}. \tag{3.81}$$

In addition, for each scenario $\omega(\xi)$, let F_ξ be an optimal dual function for the problem

$$\min\{qx \mid Wy \geq \omega(\xi) - \hat{\beta}, y \in \mathbb{Z}_+^{n_2}\}. \tag{3.82}$$

Since for any $\beta \in \mathbb{R}^{m_2}$, these functions remain dual feasible, we know from IP duality that

$$\begin{aligned} F_1(\beta) \leq \psi(\beta)\ \forall \beta \in \mathbb{R}^{m_2} \quad &\text{and} \quad F_1(\hat{\beta}) = \psi(\hat{\beta}) \ \text{ and similarly,} \\ F_\xi(\omega(\xi) - \beta) \leq \phi(\omega(\xi) - \beta)\ \forall \beta \in \mathbb{R}^{m_2} \quad &\text{and} \quad F_\xi(\omega(\xi) - \hat{\beta}) = \phi(\omega(\xi) - \hat{\beta})\ \forall \xi \in \Xi. \end{aligned}$$

Therefore, substituting ϕ and $\mathbf{E}_\xi \phi$ with F_1 and $\mathbf{E}_\xi \mathbf{F}_\xi$ in (3.67) and solving this approximate problem would yield a lower bound to (3.61) unless $\hat{\beta} = \beta^*$, in which case the optimal values would be equal. Using this fact, we now give the details of the algorithm that iteratively updates $\hat{\beta}$ and the corresponding optimal dual functions until the optimal value of (3.61) is found.

S0 Let $x^1 = \text{argmin}\{cx \mid Ax \geq b, x \in \mathbb{Z}_+^{n_2}\}$ and set $\beta^1 = Tx^1$. Initialize the dual functions list $\mathscr{F}_1 = \emptyset, \mathscr{F}_\xi = \emptyset$. Set $k := 1$.

S1 Find optimal dual functions F_1^k and $F_\xi^k, \forall \xi \in \Xi$ for the problems (3.81), (3.82) with right-hand sides set to β^k and $\omega(\xi) - \beta^k$, respectively. If

$$\max_{f_1 \in \mathscr{F}_1, f_\xi \in \mathscr{F}_\xi} \{f_1(\beta^k) + \mathbf{E}_\xi f_\xi(\omega(\xi) - \beta^k)\} \geq F_1^k(\beta^k) + \mathbf{E}_\xi F_\xi^k(\omega(\xi) - \beta^k), \tag{3.83}$$

then stop: $x^* = \text{argmin}\{cx \mid Ax \geq b, Tx \geq \beta^k, x \in \mathbb{Z}^{n_1}\}$ is an optimal solution to (3.61).

S2 Add these dual functions to the functions list: $\mathscr{F}_1 = \mathscr{F}_1 \cup F_1^k$. $\mathscr{F}_\xi = \mathscr{F}_\xi \cup_{\xi \in \Xi} F_\xi^k$. Solve the problem

$$z^k = \min_{\beta \in \mathbf{B}} \max_{f_1 \in \mathscr{F}_1, f_\xi \in \mathscr{F}_\xi} \{f_1(\beta) + \mathbf{E}_\xi f_\xi(\omega(\xi) - \beta)\} \,, \tag{3.84}$$

set β^{k+1} to an optimal solution of (3.84) and $k : k + 1$. Go to **S1**.

Note that the algorithm must terminate since B is finite. Now we show the correctness of the algorithm via the following proposition.

Proposition 3.45 *Assume that the algorithm terminates at iteration k. Then, z^{k-1} is equal to the optimal solution of (3.67), that is, $z^{k-1} = \min_{\beta \in \mathbf{B}} \psi(\beta) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta)$. Furthermore, $x^* =$ argmin$\{cx \mid Ax \geq b, Tx \geq \beta^k, x \in \mathbb{Z}^{n_1}\}$ is an optimal solution to (3.61).*

Proof. Let the function F be defined as

$$F(\beta) = \max_{f_1 \in \mathscr{F}_1, f_\xi \in \mathscr{F}_\xi} \{f_1(\beta) + \mathbf{E}_\xi f_\xi(\omega(\xi) - \beta)\} \;\; \forall \beta \in \mathbf{B}.$$

Clearly, F is a lower bounding approximation at any iteration, that is,

$$z^{k-1} = F(\beta) \leq \phi(\beta) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta) \;\; \forall \beta \in \mathbf{B} \;\;, \tag{3.85}$$

with equality at β^k at iteration k, since

$$\begin{aligned} F(\beta^k) &\geq F_1^k(\beta^k) + \mathbf{E}_\xi F_\xi^k(\omega(\xi) - \beta^k) \\ &= \phi(\beta^k) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta^k) \;\;. \end{aligned}$$

Now, assume that (3.67) attains its optimal value at β^*. Then, we must have

$$\begin{aligned} F(\beta^k) = z^{k-1} &= \phi(\beta^k) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta^k) \\ &\leq F(\beta^*) \\ &\leq \phi(\beta^*) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta^*) \\ &\leq \phi(\beta^k) + \mathbf{E}_\xi \phi(\omega(\xi) - \beta^k) \\ &= F(\beta^k) \ , \end{aligned}$$

where first inequality follows from the optimality of β^k for (3.84) at iteration $k-1$, the second inequality from (3.85), and the third inequality from the optimality of β^* for (3.67). The rest of the proof follows from Theorem 3.42. □

Observe that there are two challenges to be overcome to obtain a practical implementation of this algorithm: obtaining strong dual feasible functions and solving (3.84). The first difficulty can easily be overcome by enhancing the set $\mathscr{F}_1$ and $\mathscr{F}_\xi$ using the value functions of single-constraint approximations and generating strong dual functions from the primal solution algorithms, even for the most common solution method, branch-and-cut, as we have outlined in Section 2.2.5. Recall that strong dual functions can be obtained from the branch-and-cut procedure if the subadditive representation or the right-hand side dependency of each added cut is known.

To solve (3.84), a branch-and-bound algorithm on **B**, similar to the one given by Kong et al. [2006] for solving (3.67) can be used. Alternatively, it can be reformulated as a collection of integer programming subproblems if the generated dual functions have *nice* structures. For a given iteration and $\xi \in \Xi$, let $g_\xi(\beta) = \max_{f_\xi \in \mathscr{F}_\xi}\{f_\xi(\omega(\xi) - \beta)\}$. Then we can reformulate (3.84) as follows

$$\begin{aligned} \min \quad & cx + \sum_\xi p^\xi g_\xi(\beta) \\ \text{s.t.} \quad & Ax \geq b \\ & Tx = \beta \\ & x \in \mathbb{Z}^{r_1} \times \mathbb{R}^{n_1 - r_1} \ . \end{aligned} \tag{3.86}$$

Assume that each dual function $f_\xi \in \mathscr{F}_\xi$ is obtained from the branch-and-bound procedure and recall from Theorem 2.19 that it can be written as

$$f_\xi(\beta) = \min_{t \in T}\{v^t\beta + \underline{v}^t l^t - \overline{v}^t u^t\}\ , \tag{3.87}$$

where T is the leaf nodes of the corresponding search tree and $(v^t, \underline{v}^t, \overline{v}^t)$ is a proper dual solution for the subproblem in node $t \in T$. In this case, the reformulation reduces to an optimization problem with a piecewise-linear cost function. Therefore, it can be solved by the same approach we have used in (3.71) or (3.72) as long as we can detect the break points or the convex pieces of $\sum_\xi p^\xi g_\xi(\beta)$.

Example 22 Below, we illustrate the dual algorithm for the sample SIP problem (3.79). For the second step of each iteration, though other procedures can also be used to obtain strong dual functions at each scenario value, here we only consider the piecewise-linear concave functions delivered by our basic observations discussed before over the value function of a single-constraint case.

Iteration 1

S0 Solving the core problem $\min\{-3x_1 - 4x_2 \mid x_1 + x_2 \le 5,\ x_1 \le 5,\ x_2 \le 5,\ x_1, x_2 \in \mathbb{Z}_+\}$, we obtain $x_1^* = 0, x_2^* = 5$, and hence, $\beta^1 = \frac{5}{2}$.

S1 For the second stage problem (3.80) with right-hand side set to $6 - \frac{5}{2}$ and $12 - \frac{5}{2}$, corresponding strong dual functions, which can be obtained by branch and bound primal solution algorithm, are

$$F^1_{\omega(\xi)=6}(\beta) = \min\{3\beta - 5.5,\ -\beta + 8.5\} \quad \text{and} \quad F^1_{\omega(\xi)=12}(\beta) = \min\{3\beta - 21,\ -\beta + 17\}.$$

Then, $F(\beta) = \max\{F^1_{\omega(\xi)=6}(\beta), F^1_{\omega(\xi)=12}(\beta)\}\ \forall \beta \in \mathbb{R}$ is a lower bounding approximation of the second stage value function ϕ (see Figure 3.13).

S2 Solving

$$\begin{aligned} z^1 = \min \quad & -3x_1 - 4x_2 + 0.5(F(6-\beta) + F(12-\beta)) \\ \text{s.t.} \quad & x_1 + x_2 \leq 5 \\ & 2x_1 + \tfrac{1}{2}x_2 = \beta \\ & x_1 \leq 5, x_2 \leq 5, x_1, x_2 \in \mathbb{Z}_+ \end{aligned} \tag{3.88}$$

yields $\beta^2 = 10$ with $z^1 = -23.5$.

Iteration 2

S1 Again, we obtain dual functions for (3.80) with right-hand sides -4 and 2

$$F^2_{\omega(\xi)=6}(\beta) = \min\{3\beta + 15,\ -\beta - 1\} \text{ and } F^2_{\omega(\xi)=12}(\beta) = \min\{3\beta,\ -\beta + 8\}.$$

Since $F(-4) + F(2) < F^2_{\omega(\xi)=6}(-4) + F^2_{\omega(\xi)=12}(2)$, we continue to iterate and update the approximation with these dual functions, that is, $F = \max\{F^i_{\omega(\xi)} \mid i \in \{1,2\}, \omega(\xi) \in \{6,12\}\}$ (see Figure 3.14).

S2 Solving (3.88) with the updated F, we obtain $\beta^3 = 5.5$ with $z^2 = -16.25$.

Likewise, in iteration 3, we obtain $\beta^4 = 7.0$ with $z^3 = -15.25$ (see Figure 3.15) and in iteration 4, $\beta^5 = 5.5$ with $z^4 = -15$ (see Figure 3.16). We stop at this iteration, since $\beta^5 = \beta^3$, and the current approximation F already satisfies $F(0.5) = \phi(0.5)$ and $F(6.5) = \phi(6.5)$. □

3.4.2 Mixed Integer Bilevel Linear Programming

Consider a *mixed integer bilevel linear program* (MIBLP)

$$z_{BLP} = \max\left\{cx + q^1 y \mid x \in \mathcal{P}_U, y \in \text{ argmin } \{q^2 y \mid y \in \mathcal{P}_L(x)\}\right\}, \tag{3.89}$$

where

$$\mathcal{P}_U = \left\{x \in \mathbb{Z}^{r_1}_+ \times \mathbb{R}^{n_1 - r_1}_+ \mid A^1 x \leq b^1\right\}$$

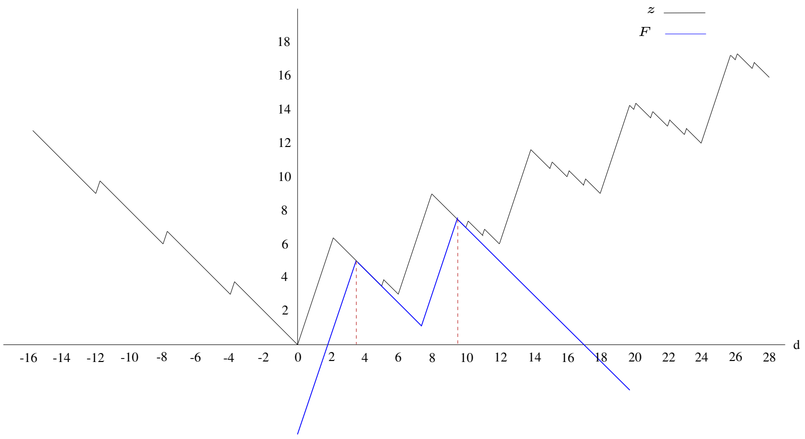


Figure 3.13: The approximate lower bounding function F obtained at iteration 1 of dual method to solve sample SIP problem (3.79).

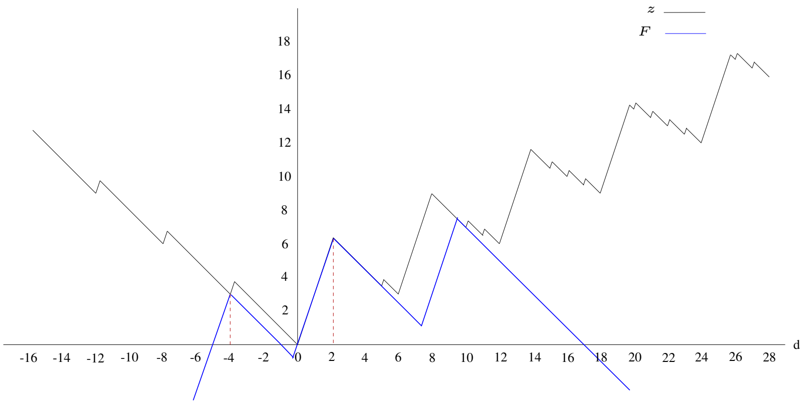


Figure 3.14: The approximate lower bounding function F obtained at iteration 2 of dual method to solve sample SIP problem (3.79).

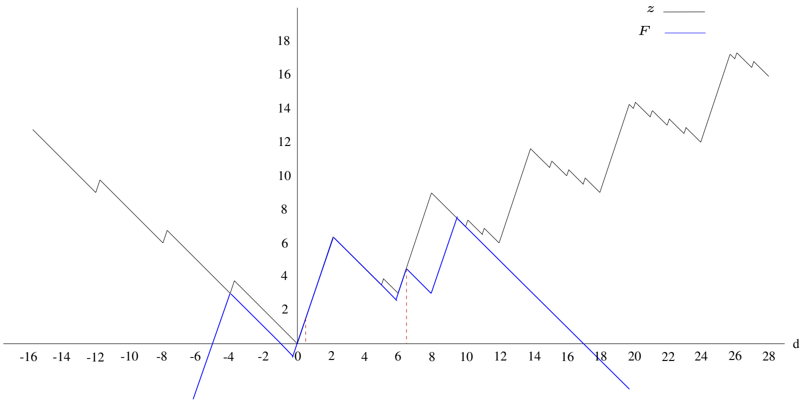


Figure 3.15: The approximate lower bounding function F obtained at iteration 3 of dual method to solve sample SIP problem (3.79).

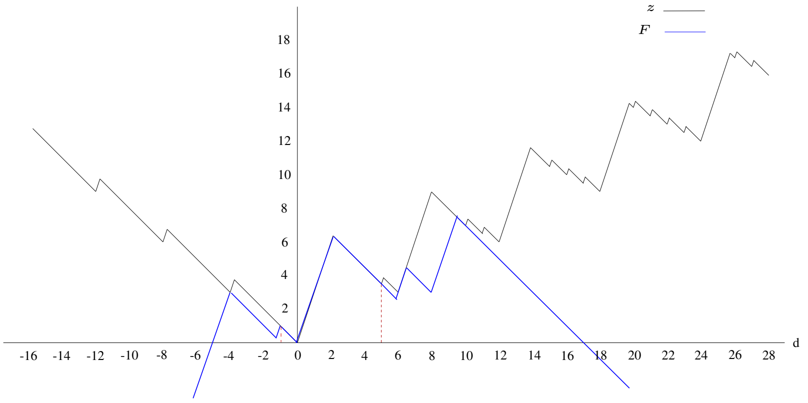


Figure 3.16: The approximate lower bounding function F obtained at iteration 4 of dual method to solve sample SIP problem (3.79).

is the *upper-level feasible region*,

$$\mathcal{P}_L(x) = \left\{ y \in \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2-r_2} \mid Wy = b^2 - A^2x \right\}$$

is the *lower-level feasible region* with respect to a given $x \in \mathbb{R}^{m_1}$ for $A^1 \in \mathbb{Q}^{m_1 \times n_1}$, $b^1 \in \mathbb{Q}^{m_1}$, $A^2 \in \mathbb{Q}^{m_2 \times n_1}$, $W \in \mathbb{Q}^{m_2 \times n_2}$, and $b^2 \in \mathbb{Q}^{m_2}$. Let $\Omega_{BLP} \equiv \{(x, y) \mid x \in P_U, y \in \mathcal{P}_L(x)\}$ and a pair $(x, y) \in \Omega_{BLP}$ is called *bilevel feasible*, if $y \in$ argmin $\{q^2y \mid y \in \mathcal{P}_L(x)\}$. We assume that $\Omega_{BLP} \neq \emptyset$, $\mathcal{P}_U$ is finite and $\mathcal{P}_L(x) \neq \emptyset \; \forall x \in \mathcal{P}_U$. Furthermore, consider the value function

$$z_L(\alpha) = \min\{q^2y \mid Wy = \alpha, y \in \mathbb{Z}^{r_2} \times \mathbb{R}^{n_2-r_2}\} \;\; \forall \alpha \in \mathbb{R}^{m_2}. \tag{3.90}$$

Then, we can reformulate the MIBLP (3.89) as follows

$$\begin{array}{ll} \max & cx + q^1y \\ \text{subject to} & A^1x \leq b^1 \\ & A^2x + Wy = b^2 \\ & q^2y = z_L(b^2 - A^2x) \\ & x \in \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1-r_1}, y \in \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2-r_2}\,. \end{array} \tag{3.91}$$

With arguments similar to those in the previous section, we can derive an algorithm to solve (3.90) by iteratively updating an approximation of the value function. We illustrate the dual solution algorithm with an upper bounding approximation.

S0 Let (x^1, y^1) be the optimal solution to (3.91) without the value function constraint and set $k := 1$.

S1 Solve (3.90) for $\alpha^k = b^2 - A^2x^k$. If $q^2y^k = z_L(\alpha^k)$, then stop. (x^k, y^k) is an optimal solution to MIBLP problem (3.89).

S2 Let G^k be the strong upper bounding function obtained by Theorem 3.39 for α^k and solve

$$\begin{aligned} z^k = \max \quad & cx + q^1 y \\ \text{subject to} \quad & A^1 x \leq b^1 \\ & A^2 x + Wy = b^2 \\ & q^2 y \leq G^i(b^2 - A^2 x) \quad i = 1, \ldots, k \\ & x \in \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}, y \in \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2} \ . \end{aligned} \tag{3.92}$$

Let (x^{k+1}, y^{k+1}) be an optimal solution to (3.92), set $k : k+1$ and go to **S1**.

Our first observation is that $z^i \geq z^{i+1}$ for iterations i and $i+1$, since the subproblem (3.92) solved at iteration i is a relaxation of the subproblem at $i+1$. In addition, if $i+1$ is not the last iteration, then $x^i \neq x^{i+1}$. Note that the constraint generated from the strong upper bounding function at $b^2 - A^2 x^i$ and added at iteration i guarantees $q^2 y^{i+1} \leq z_L(b^2 - A^2 x^i)$ and if $x^{i+1} = x^i$, then we would have $q^2 y^{i+1} = z_L(b^2 - A^2 x^i) = z_L(b^2 - A^2 x^{i+1})$ which contradicts that $i+1$ is an intermediate iteration. Consequently, the algorithm terminates after finitely many steps, since $\mathcal{P}_U$ is finite. Now we show the correctness of the algorithm via the following proposition.

Proposition 3.46 *Assume that the algorithm terminates at iteration k. Then (x^k, y^k) is an optimal solution to (3.89).*

Proof. Observe the last subproblem (3.92) we solve at iteration k:

$$\begin{aligned} z^k = \max \quad & cx + q^1 y \\ \text{subject to} \quad & A^1 x \leq b^1 \\ & A^2 x + Wy = b^2 \\ & q^2 y \leq G(b^2 - A^2 x) \\ & x \in \mathbb{Z}_+^{r_1} \times \mathbb{R}_+^{n_1 - r_1}, y \in \mathbb{Z}_+^{r_2} \times \mathbb{R}_+^{n_2 - r_2} \ , \end{aligned} \tag{3.93}$$

where $G(\alpha) = \min_i \{G^i(\alpha)\} \ \forall \alpha \in \mathbb{R}^{m_2}$. Now, assume that we replace $q^2 y \leq G(b^2 - A^2 x)$ in (3.93) with $q^2 y = z_L(b^2 - A^2 x)$, which results in the MIBLP formulation (3.91). Note that

(x^k, y^k) is still a feasible solution, since $q^2 y^k = z_L(b^2 - A^2 x^k)$ due to termination at iteration k. Furthermore, it is also an optimal solution because (3.93) is a relaxation of (3.91) since G is an upper bounding function: $G(\alpha) \geq z_L(\alpha) \;\; \forall \alpha \in \mathbb{R}^{m_2}$. □

If the upper bounding functions added at last step of each iteration is obtained from Corollary 3.40, then the subproblem (3.93) reduces to a mixed integer linear program with a piecewise-linear constraint function since the upper bounding function obtained by (3.59) is piecewise-linear concave itself. In this case, each subproblem can be treated as in our discussion of the dual solution method given for stochastic programs.

Alternatively, we can embed the above dual method into either a cutting plane or a branch-and-bound framework using the structure of the upper bounding function (3.59). To be complementary with Corollary 3.40, let I_2 and C_2 be the set of indices of integer and continuous variables in the lower level and assume that the dual feasible polyhedron of the continuous restriction of the lower level problem

$$\{u \in \mathbb{R}^{m_2} \mid uW_{C_2} \leq q^2_{C_2}\} \tag{3.94}$$

is nonempty and bounded. Then for a fixed x^*, the strong upper bounding function on z_L obtained at $b^2 - A^2 x^*$ by Corollary 3.40 is

$$G(\alpha) = q^2_{I_2} y^*_{I_2} + \max_{v \in V_L}\{v(\alpha - W_{I_2} y^*_{I_2})\} \; \forall \alpha \in \mathbb{R}^{m_2} \; , \tag{3.95}$$

where y^* is an optimal solution for $\min\{q^2 y \mid y \in \mathcal{P}_L(x^*)\}$ and V_L is the set of extreme points of dual feasible polytope (3.94). For each $v \in V_L$, let $\Omega_v \equiv \{\alpha \in \mathbb{R}^{m_2} \mid G(\alpha) = q^2_{I_2} y^*_{I_2} + v(\alpha - W_{I_2} y^*_{I_2})\}$. Setting $V_L = \{v_1, \ldots, v_R\}$, we obtain the valid disjunctions

$$\begin{array}{c} \alpha \in \Omega_{v_1} \\ q^2 y \leq q^2_{I_2} y^*_{I_2} + v_1(\alpha - W_{I_2} y^*_{I_2}) \end{array} \quad \text{OR} \;\; \ldots \;\; \text{OR} \quad \begin{array}{c} \alpha \in \Omega_{v_R} \\ q^2 y \leq q^2_{I_2} y^*_{I_2} + v_R(\alpha - W_{I_2} y^*_{I_2}) \end{array} \tag{3.96}$$

These disjunctions, in turn, can either be used to generate a disjunctive valid inequality to cut off the bilevel feasible solution (x^*, y^*) or can be applied as a branching rule in a branch-and-bound algorithm to partition the subproblem (3.93) into a set of mixed integer linear problems. We illustrate the latter approach in the following example.

Example 23 Consider the MIBLP instance

$$\begin{aligned} z_{BLP} = \max \quad & 3x_1 + 4x_2 + y_1 + y_2 - y_3 - y_4 - 7y_5 \\ \text{s.t.} \quad & x_1 + x_2 \leq 6 \\ & x_1 \leq 5, \ x_2 \leq 5 \\ & x_1, x_2 \in \mathbb{Z}_+ \\ & y \in \text{argmin}\{ \quad 3y_1 + \tfrac{7}{2}y_2 + 3y_3 + 6y_4 + 7y_5 \\ & \qquad\qquad \text{s.t.} \quad 6y_1 + 5y_2 - 4y_3 + 2y_4 - 7y_5 = 12 - 2x_1 - \tfrac{7}{2}x_2 \\ & \qquad\qquad\qquad y_1, y_2, y_3 \in \mathbb{Z}_+, y_4, y_5 \in \mathbb{R}_+ \ \}. \end{aligned} \tag{3.97}$$

Clearly, the value function of the lower level problem z_L defined by (3.90) is the same as the value function of our running instance in Example 8 and is given in Figure 3.1.

Initially, we solve the problem (3.91) without the value function constraint, that is,

$$\begin{aligned} \max \quad & 3x_1 + 4x_2 + y_1 + y_2 - y_3 - y_4 - 7y_5 \\ \text{s.t.} \quad & x_1 + x_2 \leq 6 \\ & x_1 \leq 5, \ x_2 \leq 5 \\ & 6y_1 + 5y_2 - 4y_3 + 2y_4 - 7y_5 = \alpha \\ & \alpha = 12 - 2x_1 - \tfrac{7}{2}x_2 \\ & x_1, x_2, y_1, y_2, y_3 \in \mathbb{Z}_+, y_4, y_5 \in \mathbb{R}_+ \ , \end{aligned} \tag{3.98}$$

and obtain the optimal solution $x^0 = (1, 5)$ with $y^0 = (0, 0, 2, \frac{1}{4}, 0)$. At $\alpha^0 = b^2 - A^2x^0 = 12 - 2*1 - \frac{7}{2}*5 = -7.5$, we have $z_L(-7.5) = 6.5$. Since $z_L(-7.5) < q^2y^2 = 2*3 + \frac{1}{4}*6 = 7.5$, we begin iteratively approximating z_L with the upper level bounding function obtained by Corollary 3.40 at α^0:

$$G^0(\alpha) = 3 + \max\{3(\alpha + 4), -(\alpha + 4)\} \ \ \forall \alpha \in \mathbb{R} \ , \tag{3.99}$$

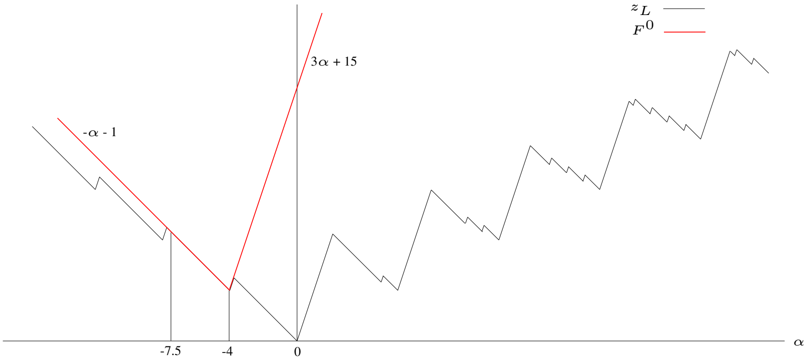


Figure 3.17: The strong upper bounding function G^0 at $\alpha^0 = -7.5$ obtained for the lower level value function z_L of sample MIBLP problem (3.97).

since $y^*_{I_2} = (0, 0, 1)$ is the integral part of an optimal solution that yields $z_L(-7.5)$. Now, we need to solve (3.98) with the following additional constraint

$$q^2 y \leq G^0(\alpha), \tag{3.100}$$

where $q^2 y = 3y_1 + \frac{7}{2}y_2 + 3y_3 + 6y_4 + 7y_5$. As we discussed above, G^0 is separable at $\alpha = b^2 - A^2x^0 - W_{I_2}y^*_{I_2} = -4$, and (3.98) can be decomposed into two subproblems applying the following disjunction as a branching rule (see Figure 3.17):

$$\begin{array}{ccc} \alpha \leq -4 & & \alpha \geq -4 \\ & \text{OR} & \\ q^2 y \leq -\alpha - 1 & & q^2 y \leq 3\alpha + 15. \end{array}$$

For the first subproblem, we obtain $x^1 = (3, 3)$ and $y^1 = (0, 0, 1, 0, \frac{1}{14})$. At $\alpha^1 = b^2 - A^2x^1 = -4.5$, $z_L(-4.5) = 3.5$ is attained by y^1, i.e., $z_L(-4.5) = q^2y^1$, and hence, (x^1, y^1) is bilevel feasible and constitutes an incumbent solution with value $z^1_{BLP} = 19.5$. Note that we do not need to further iterate on this subproblem since replacing the branching rule with

$$\begin{array}{c} \alpha \leq -4 \\ q^2 y = z_L(\alpha) \end{array}$$

would give the same result following our discussion for the proof of Proposition 3.46. However, if (x^1, y^1) was not bilevel feasible, we would keep iterating/branching for this subproblem after obtaining the corresponding strong upper bounding function at α^1. Similarly, we get the solution $x^2 = (5, 1), y^2 = (0, 0, 0, 0, \frac{3}{14})$ for the second subproblem. At $\alpha^2 = -1.5$, $z_L(-1.5) = 1.5$ is also attained by y^2 and hence, (x^2, y^2) is again bilevel feasible with solution value $z^2_{BLP} = 17.5$. Then, we conclude that (x^1, y^1) is an optimal solution of (3.97) with $z_{BLP} = 19.5$. □

Chapter 4

Warm Starting and Sensitivity Analysis

Although we usually assume the input to a mathematical program is known and deterministic, one often wants to study the effect of small perturbations to the input on the outcome in order to determine how sensitive the decision made is to such perturbation. For instance, it may be desirable to determine how the optimum may change if available resources change unexpectedly. For such purposes, the tools of post-solution analysis are appropriate.

Post-solution analysis mainly addresses the following questions:

- *Local sensitivity analysis:* Assume that a given mathematical program is solved to optimality. Is the current solution still optimal after a given change in the data? Or, more generally, for what ranges of problem data does the current solution remain optimal?

- *Warm Starting:* If the current solution is not optimal, can we make use of the information collected through the solution process to get or estimate the optimal solution of the modified problem?

In the LP case, these questions are well-studied and tools for these types of analyses well-developed. The optimality conditions for LP allow us to easily verify whether the current solution is still optimal for a modified problem or to determine the ranges over which the optimality conditions still hold. In case the conditions do not hold, rather than solving the new problem from scratch, the current or a slightly modified basis can be used to hot start the simplex method in order to more quckly get the new optimal solution (see relevant sections of Bertsimas and Tsitsiklis [1997]).

The importance of post-solution analysis for MILPs is clear. Because of the lack of an efficient algorithm for solving MILPs, it may not be practical to solve a modified problem from scratch. Post-solution analysis may reveal useful information and yield a better starting point for a potential re-solve. However, the functionality of these tools for MILPs is very limited compared to the techniques introduced for LPs. First of all, there is no relevant concept of primal-dual algorithm comparable to the simplex method. Llewellyn and Ryan [1993] discuss a convergent primal-dual algorithm for PILP problems but it consists of *black boxes* that require the solution of some PILPs that may be as hard to solve as the original problem. Second, the sufficient conditions that can be applied may be too weak to determine the optimality of the current solution for the modified problem and aside from that, are not necessary in general. In other words, even if the current solution does remain optimal for the modified problem, existing methods may not be able to detect it. In fact, for some cases even determining the range of a single parameter in which an incumbent solution remains optimal is difficult (Schulz [2009]). Third, there are substantial computational difficulties to be overcome. Note that in the LP case, all relevant the information is restored in and the analysis results are just the optimal basis, but in MILP case, one may need to keep track of each single iteration and save a lot of information including the useless ones.

In the following sections, we will outline existing methods to partially answer the questions above and provide extensions based on the theory in Chapters 2 and 3. The implementation of these methods depends implicitly on the solution algorithm of the base instance. Therefore, although there are some post-solution analysis studies for other algorithms, such as implicit enumeration (Roodman [1972], Loukakis and Muhlemann [1984]) and Lagrangian relaxation (Shapiro [1977]), we will pay most of our attention to branch-and-bound and branch-and-cut algorithms. Also note that some special cases, such as knapsack (Blair [1998]), general assignment problems (Nauss [1979]) and other combinatorial problems(Libura [1991, 1996], Sotskov et al. [1995]) have been studied more extensively. However, our main concern here is dual approaches for general

MILPs.

4.1 Dual Approach

The primal solution algorithms for MILPs can be viewed as iterative procedures to reach optimality conditions by updating and closing the *duality gap* which is the measure of *distance from optimality*. The duality gap is updated by means of a lower and an upper bound from the primal and dual information obtained so far and at each iteration, additional restrictions such as new cutting planes are added or new partitions, such as in branch-and-bound algorithm, are formed in order to attain the optimality by reducing the duality gap.

The dual approach we are going to construct for sensitivity analyses and warm-starting for the rest of the chapter relies on the duality results discussed before. Recall that primal solution algorithms generate information not only for solving the given instance but also for approximating the value function of the primal instance through dual functions. These dual functions, in turn, can be used either to estimate the value function for other right-hand sides or to initiate the primal solution algorithm from an advanced starting point to solve a modified version of the primal instance.

To be more specific, assume that we use the branch-and-bound algorithm to solve the primal problem (1.7) with fixed right-hand side b. We solve the LP relaxation of the following subproblem at each node t of the branch-and-bound tree:

$$z_t(b) = \min_{x \in \mathcal{S}_t(b)} \quad cx \tag{4.1}$$

where $\mathcal{S}_t(b) = \{x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+ \mid Ax \geq b, x \geq l^t, -x \geq u^t\}$. Let T_L be the set of leaf nodes of current tree T and for each $t \in T_L$, $(v^t, \underline{v}^t, \overline{v}^t)$, be the corresponding dual solution (as defined in Section 2.2.5 without cutting planes) for the LP relaxation of (4.1). As we have discussed before, the dual function, or the lower bounding function on the value function, that can be obtained from the tree T is

$$F_T(d) = \min_{t \in T_L} \{v^t d + \underline{v}^t l^t - \overline{v}^t u^t\} \quad \forall d \in \mathbb{R}^m. \tag{4.2}$$

Similarly, we can extract an upper bounding function considering the nodes that yield feasible solutions to the initial primal problem. In particular, let for a given $d \in \mathbb{R}^m$, $T_U(d) \subseteq T$ be such that $t \in T_U(d)$ if the current basis of node t yields a basic solution $x^t \in \mathcal{S}_t(d) \cap \mathcal{S}(d)$. Then the upper bounding function $G_T : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{\infty\}$ on the value function can be written as

$$G_T(d) = \begin{cases} \min\limits_{t \in T_U(d)} \{cx^t\} & \text{if } T_U(d) \neq \emptyset, \\ \infty & \text{otherwise.} \end{cases} \tag{4.3}$$

Clearly, we have

$$G_T(d) \geq z(d) \geq F_T(d) \; \forall d \in \mathbb{R}^m. \tag{4.4}$$

For the fixed primal problem (1.7), the dual gap at T is measured by the difference between $G_T(b)$ and $F_T(b)$ and the algorithm iterates until it reaches a tree T^* satisfying $F_{T^*}(b) = G_{T^*}(b)$. In this sense, the branch-and-bound algorithm can be seen as an iterative algorithm that constructs upper and lower bounding functions, and terminates as soon as the bounding functions approximate the value function closely enough at a given right-hand side.

Sensitivity analysis in our framework is simply the study of the behavior of these bounding functions. Using these functions, we can state *sufficient* conditions on the optimality of a modified instance by duality results and by checking the proximity of the bounding functions to each other. For changes in the right-hand side, for instance, this is done by checking (4.4). Otherwise, we obtain bounds on the optimal value of the modified instance.

On the other hand, since the branch-and-bound solution algorithm produces bounding functions, an existing tree and the corresponding bounding functions can be modified and fed into the solution procedure of a modified instance to start the algorithm from an advanced level. This method is the basis for warm-starting and the idea is that if the solution procedure can be initiated from existing bounding functions, then the duality gap at a modified right-hand side might be closed quicker than computing from scratch.

Note that the *quality* of the bounding functions, and hence the behavior of sensitivity analysis and warm-starting, entirely depend on the disjunctions branching decisions made and the resulting partition of the feasible region. Therefore, different branching decisions and using different subtrees of current branch-and-bound tree to obtain the bounding functions might completely change the quality of the results of these techniques. In addition, although we have discussed here the particular case with branch-and-bound algorithm and changes on right-hand side, as it will be apparent later in this chapter, the basic principals of the dual approaches for sensitivity analysis and warm-starting outlined above are still valid and can be applied for other primal solution procedures like branch-and-cut and other modifications of the primal instance such as changes to the objective coefficients.

4.2 Sensitivity Analysis

Note from (1.7) that the input data for a general MILP can be defined by the quadruple (A, b, c, r). Assuming that we have modified the input data, we first partially answer the question of how to carry out sensitivity analysis after some or all of these modifications through a given strong dual function. Then we discuss sufficient conditions for optimality through dual functions appropriate for each type of modification and obtained from a primal solution algorithm.

Basic Observations. Assume that both the original and the modified problems are feasible and let (x^*, F^*) be an optimal primal- subadditive dual solution pair for (A, b, c, r). The following basic observations (Geoffrion and Nauss [1977], Wolsey [1981]) state the conditions under which primal feasibility, dual feasibility and optimality still hold for x^* and F^* after changes made to the input. They are derived for PILP problems but can be easily adapted to MILP case.

1. $(A, b, c) \rightarrow (A, \tilde{b}, c)$

    1a. F^* remains dual feasible due to the nature of dual functions.

    1b. Let $Y^* = \{y \mid F^*(Ay) = cy\}$. If F^* remains optimal, then the new optimal solution is in Y^* since, given $z(\tilde{b}) = c\tilde{x}^*$, $c\tilde{x}^* = F^*(\tilde{b}) = F^*(A\tilde{x}^*)$.

2. $(A, b, c) \rightarrow (A, b, \tilde{c})$

2a. x^* remains primal feasible.

2b. If $F^*(a_j) \leq \tilde{c}_j$ for all j, then F^* remains dual feasible.

2c. If $F^*(a_j) \leq \tilde{c}_j$ when $x^*_j = 0$ and $\tilde{c}_j = c_j$ otherwise, then x^* remains optimal since F^* remains dual feasible by part (2b) and

$$\tilde{c}x^* = \sum_{j,x^*_j>0} \tilde{c}_j x^*_j + \sum_{j,x^*_j=0} \tilde{c}_j x^*_j = \sum_{j,x^*_j>0} c_j x^*_j + \sum_{j,x^*_j=0} c_j x^*_j = cx^* = F^*(b). \quad (4.5)$$

2d. If $c_j \leq \tilde{c}_j$ when $x^*_j = 0$ and $\tilde{c}_j = c_j$ otherwise, then x^* remains optimal. This is clear from part (2c) and gives sensitivity information independent of the optimal dual function.

2e. Let the vector τ denote the upper bounds on the variables. If $\tilde{c}_j \leq c_j$ when $x^*_j = \tau^*_j$ and $\tilde{c}_j = c_j$ otherwise, then x^* remains optimal. Consider the problem

$$\begin{aligned} \min\{\tilde{c}x + \tau(c - \tilde{c}) \mid x \in \mathcal{S}(b)\} &= \min\{\tilde{c}x + cx - cx + \tau(c - \tilde{c}) \mid x \in \mathcal{S}(b)\} \\ &= \min\{cx + (c - \tilde{c})(\tau - x) \mid x \in \mathcal{S}(b)\} \end{aligned}$$

Note that the optimal solution to this problem is x^* since $(c - \tilde{c})(\tau - x) \geq 0$ and $cx^* \leq cx$ for all $x \in \mathcal{S}(b)$.

3. $(A, b, c) \rightarrow (\tilde{A}, \tilde{b}, \tilde{c})$

3a. If the original problem is a relaxation of the new problem, then x^* remains optimal.

3b. When a new activity $(\tilde{c}_{n+1}, \tilde{a}_{n+1})$ is introduced, $(x^*, 0)$ remains feasible.

3c. Furthermore, if $F^*(\tilde{a}_{n+1}) \leq \tilde{c}_{n+1}$, then $(x^*, 0)$ remains optimal since F^* remains dual feasible and $cx* = F^*(b)$.

3d. When a new constraint $(\tilde{a}^{m+1})$ with right-hand side $\tilde{b}_{m+1}$ is introduced, if x^* is still feasible, then it remains optimal by part (3a).

3e. Furthermore, let the function $\tilde{F} : \mathbb{R}^{m+1} \rightarrow \mathbb{R}^m$ be defined by $\tilde{F}(d, d_{m+1}) = F^*(d)$.

Then $\tilde{F}$ is dual feasible for the new problem since $\tilde{F}$ is subadditive and $\tilde{F}(a_j, \tilde{a}_j^{m+1}) = F^*(a_j) \leq c_j,\ j \in I$.

Also note that if one drops the subadditivity requirement of F^*, then (2b), (2c) and (3c) are no longer valid.

Next, we study each case separately either to derive sufficient conditions to check optimality or to extract bounds on the optimal solution value for a modified problem using the information collected from a primal solution algorithm. As outlined in previous section, we derive these methods over dual functions. Note that we have already discussed in Sections 2.2.1 and 2.2.5 these type of dual functions obtained from the last iteration (for cutting-plane algorithm) or the leaf nodes of the resulting tree (for branch-and-cut algorithm) of the corresponding primal solution algorithm. However, for the purpose of a sensitivity analysis, we can sacrifice more computational effort and more space to keep all dual information from all stages of the primal solution algorithm to extract a dual function to better approximate the value function.

Modification to the right-hand side. In this section, we analyze the effects of modifying the right-hand side of the primal instance (1.7). For simplicity, we assume that (1.7) is a pure integer program. We give the details of extending the following results to MILP case unless it is straightforward.

Klein and Holm [1979] give sufficient conditions for optimality when the cutting plane algorithm with Gomory fractional cuts is used. In particular, when the algorithm terminates, the optimal basis of the last LP is used to check the optimality of the solution for the new right-hand side, since it remains dual feasible for the modified problem with the cuts modified appropriately (see Section 2.2.1). To describe this notion of sensitivity analysis in our framework for PILPs, let F_{CP}^i be the dual function (2.26) extracted at iteration i, $\sigma_i : \mathbb{R}^m \to \mathbb{R}^{m+i-1}$ be the recursive representation (as defined for (2.26) of the dependency of the right-hand side vector of the added cutting planes on the right-hand side of the original constraints and B^i be the corresponding set of indices of variables in the current optimal basis. Then we define the lower and upper bounding approximations as follows:

$$F_{CP}(d) = \max_i \{F^i_{CP}(d)\} \ ,$$
$$G_{CP}(d) = \begin{cases} \min\limits_i \{c_{B^i}(B^i)^{-1}\sigma_i(d)\} & \text{if}(B^i)^{-1}\sigma_i(d) \geq 0 \text{ and integer for some } i \\ \infty & \text{otherwise.} \end{cases}$$

The characteristics of these functions are easy to understand from the properties of dual functions and LP duality. F_{CP} is a dual function and yields a better approximation of the value function than that obtained bu considering each F^i_{CP} separately. On the other hand, G_{CP} ensures the primal feasibility of the basis of some iteration i and hence is an upper bounding approximation of the value function z. Consequently, we can state the sufficient conditions for optimality for the PILP instance with a modified right-hand side with the following proposition.

Proposition 4.1 *For a given $\tilde{b} \in \mathbb{R}^m$, if $F_{CP}(\tilde{b}) = G_{CP}(\tilde{b})$, then $z(\tilde{b}) = F_{CP}(\tilde{b})$.*

If the test of Proposition 4.1 fails, we still get bounds on the optimal solution value of the modified instance, that is, $G_{CP}(\tilde{b}) \geq z(\tilde{b}) \geq F_{CP}(\tilde{b})$. These bounds, in turn, can be used as a measure of closeness to optimality.

A similar result for branch-and-bound(or branch-and-cut) framework can be derived as well in terms of dual feasible bases of LP subproblems of tree nodes. Note that we have so far only used the information from the leaf nodes of the branch-and-bound tree in order to obtain a dual function (see Sections 2.2.5 and 4.1). However, this scheme can be improved by considering the vast amount of dual information revealed during processing of the intermediate nodes.

i. A dual function can be extracted from the leaf nodes of some subtrees of the given branch-and-bound tree. Let $\mathscr{T}$ be the set of all subtrees of T such that $K \in \mathscr{T}$ if and only if K is connected and rooted at the root node of T and both left and right children of an intermediate node $t \in K$ are also in K. Furthermore, for $K \in \mathscr{T}$, let K_L be the set of leaf nodes of subtree K. Then the dual function (4.2) can be strengthen as follows

$$F(d) = \max_{K \in \mathscr{T}} \min_{t \in K_L} \{v^t d + \underline{v}^t l^t - \overline{v}^t u^t\} \quad \forall d \in \mathbb{R}^m. \tag{4.6}$$

ii. Assuming the variables have both lower and upper bounds initially, then the dual solution of each node is also a dual solution of the other nodes of the tree. Then the dual function (4.6) can be strengthen as follows

$$F(d) = \max_{K \in \mathscr{T}} \min_{t \in K_L} \max_{s \in T} \{v^s d + \underline{v}^s l^t - \overline{v}^s u^t\} \quad \forall d \in \mathbb{R}^m. \tag{4.7}$$

Schrage and Wolsey [1985] give a recursive algorithm to evaluate (4.6) considering the relation between an intermediate node and its offsprings. For a $K \subset \mathscr{T}$, let $\bar{t} \in K_L$ be such that $L(\bar{t}), R(\bar{t}) \in T$ where $L(\bar{t})$ and $R(\bar{t})$ are the left and right children of $\bar{t}$ and consider the dual functions F_K, $F_{\bar{K}}$ obtained from the subtrees K and $\bar{K} = K \cup \{L(\bar{t}), R(\bar{t})\}$ by the procedure (4.2). Also, let us define

$$\varpi^t(d) = v^t d + \underline{v}^t l^t - \overline{v}^t u^t \quad \forall d \in \mathbb{R}^m$$

for all $t \in T$. Then for $d \in \mathbb{R}^m$, we clearly have

$$\max\{F_K(d), F_{\bar{K}}(d)\} = \begin{cases} F_K(d) & \text{if } \varpi^t(d) \geq \min\{\varpi^{L(t)}, \varpi^{R(t)}\} \\ F_{\bar{K}}(d) & \text{otherwise.} \end{cases} \tag{4.8}$$

Theorem 4.2 *Schrage and Wolsey [1985] For each node t $\in T$, let the function κ^t be defined by*

i. $\kappa^t(d) = \varpi^t(d)$ *if t is a leaf node.*

ii. $\kappa^t(d) = \max\{\varpi^t(d),\ \min\{\kappa^{L(t)}(d),\ \kappa^{R(t)}(d)\}\}$ *if t is not a leaf node and* $L(t), R(t)$ *are the indices of t's offsprings.*

Also, let the index of root node be 0*. Then,* $F(d) = \kappa^0(d)\ \forall d \in \mathbb{R}^m$ *where* F *is defined by (4.6).*

Proof. The proof follows from the relation (4.8), the inductive property of κ and from the structure of the branch-and-bound tree. □

On the other hand, the dual function (4.7) can be obtained by Theorem 4.2 after replacing the definition of ϖ^t considering the dual solutions of other nodes, i.e., by letting

$$\varpi^t(d) = \max_{s \in T}\{v^s d + \underline{v}^s l^t - \overline{v}^s u^t\} \ \ \forall d \in \mathbb{R}^m. \tag{4.9}$$

In this case, evaluating κ^0, and hence the dual function (4.7), might be expensive due to the size of T. One possible approach to achieving a computationally feasible scheme is to consider in (4.9) only a subset of T. However, it is not clear how to determine such a subset so that the resulting dual function would still approximate the value function as closely as when using the set T.

We can extend the above analyses to branch-and-cut by the same procedure we applied to obtain a dual function from a branch-and-cut tree (see Section 2.2.5). We simply need to modify our definition of ϖ^t for each node t by adding the dual information coming from the generated cuts. To see this, assume that the problem is solved with the branch-and-cut algorithm and that the subadditive representation or the original right-hand side dependency of each cut is known. Then the analog of dual function (4.6) can be obtained from Theorem 4.2 by setting

$$\varpi^t(d) = v^t d + \underline{v}^t l^t - \overline{v}^t u^t + \sum_{k=1}^{\nu(t)} w^t{}_k F^t_k(\sigma_k(d)) \ \ \forall d \in \mathbb{R}^m, \tag{4.10}$$

where $\nu(t)$ is the number of total cuts added, F^t_i is the right-hand side dependency of cut i on the original right-hand side and σ_i is as defined in (2.2.1). If we do not know the original right-hand side dependencies of the cuts, we can still use the variable bounds and set for each node t

$$\varpi^t(d) = u^t d + \underline{u}^t l^t - \overline{u}^t g^t + \max\{w^t \tilde{h}^t,\ w^t \hat{h}^t\} \ \ \forall d \in \mathbb{R}^m, \tag{4.11}$$

where $\tilde{h}^t$ and $\hat{h}^t$ are defined as in Theorem 2.21.

Once we extract an upper bounding function (4.3) and a dual function (4.6) from a branch-and-bound (or branch-and-cut) primal solution algorithm, then we can easily state sufficient conditions for optimality for a modified instance with a new right-hand side just like Proposition 4.1.

Modification to the objective function. We have so far throughout the thesis only discussed the value function obtained by considering the optimal value of the primal problem as a function of the right-hand side and constructed our notion of duality based on this function. However, it is also possible to define a similar function as a function of the objective coefficients and derive similar/analogous results for this case. The *objective value function* can be written as

$$Z(q) = \min_{x \in \mathcal{S}} qx \quad \forall q \in \mathbb{R}^n, \tag{4.12}$$

and for the sensitivity analysis purposes, we would be interested in deriving lower and upper bounding functions F and G from the primal solution algorithms satisfying

$$F(q) \leq Z(q) \leq G(q) \quad \forall q \in \mathbb{R}^n. \tag{4.13}$$

Note that for any objective function q, the feasible solutions found during the branch-and-bound (or branch-and-cut) algorithm remain primal feasible and an upper bounding function analogous to (4.3) can be derived easily. However, the optimal bases of the leaf nodes of the branch-and-bound tree T might not remain dual feasible for their LP subproblems and therefore, a valid lower bounding function cannot be obtained directly. For a given $q \in \mathbb{R}^n$ however, one possible way to obtain a valid lower bound is to re-optimize the LP subproblem of each leaf node t to obtain corresponding optimal feasible solutions. In particular, for $t \in T_L$, let x^t be an optimal solution for the LP subproblem (4.1) with objective coefficients q. Then

$$F(q) = \min_{t \in T_L} \{qx^t\} \tag{4.14}$$

would give a lower bound for the value $Z(q)$. With these bounds, one can check the optimality of the modified problem by querying sufficient optimality conditions just as stated for cutting-plane method for right-hand side analysis through Proposition 4.1.

Other modifications. For any modification, the same procedure applies. Note that in many cases, the current bases of LP subproblems can be extended to dual feasible bases with none to a small effort. For instance, if a new constraint added a dual feasible basis for each node is readily available. Then, in a similar fashion, the Propositions 4.1 can be extended to this specific case. However, if multiple changes are allowed, it may be hard to obtain the dual bases or re-optimize the LP subproblems.

4.3 Warm Starting

In mathematical programming, it is common in practice that one needs to re-solve a modified problem or to solve a series of related MILPs. Such sequences of problems arise both in dealing with uncertainty and when implementing decomposition techniques (see Section 4.5). The approach most authors have taken in such circumstances is to solve each of those problems from scratch. This approach is very straightforward, but ignores the fact that the solution process of one instance may reveal useful information that allows the solution process of another instance to begin from an advanced level.

Warm starting is a technique to collect information during the solution procedure of one instance in order to make use of this information to initiate the solution procedure for a related problem. From the view point of duality, warm-starting information can be seen as a method of using predetermined bounding functions as input in order to that allow the solution algorithm to more quickly solve a modified instance. Here, by 'solving', we mean to update/strengthen these bounding functions at each iteration of solution procedure until optimality is reached, where optimality is determined by the distance between these two functions at some point (Ralphs and Güzelsoy [2006]).In the LP case, if the simplex method is used, warm-starting information consists of the optimal basis of the original instance. For the right-hand side value function, the corresponding warm-starting information is a dual function that is strong with respect to an initial right-hand side (see Figure 1.1). Similarly, we define the warm-starting information for an MILP solution algorithm with respect to the components of the primal solution algorithm that yielded the bounding functions. In the branch-and-cut algorithm, for instance, it refers to a given subtree, final basis of each node and enough information to modify each cut to be valid for a modified problem (see Section 2.2.5).

For the cutting plane algorithm, Feautrier [1988] discusses of using parameterized Gomory cuts with parameterized dual simplex as warm-starting information to find the integral lexicographic minimum of a parametric polyhedron. For the branch-and-bound algorithm, Geoffrion and Nauss [1977] describes three ways to take advantage of warm-starting information:

a. If a sequence of problems are to be solved, expand the solution algorithm of each instance in a manner that will provide additional information for the solution of other subsequent problems. For instance, when each problem is optimized over the same feasible region, Piper and Zoltners [1976] propose for binary PILPs to modify the branch-and-bound algorithm such that it will terminate only after finding the K best feasible solutions, hoping that one of these solutions will be optimal for another subsequent problem. The disadvantage of this improvement is that it may require diving a lot deeper in the tree and may not be efficient for general PILPs or MILPs.

b. Generalize constructing the branch-and-bound tree from considering a single instance to a family of related instances. This requires either consideration of all family members at each node, applying the same cuts and branching and terminating after all members have been solved to optimality, or consideration of only one of the instances at each node but a change in the rule for fathoming a node such that a node is not fathomed until it is guaranteed not to yield a better solution for all other instances. Recall that the methods of Rountree and Gillett [1982], Marsten and Morin [1977], Ohtake and Nishida [1985] for parametric programming are of this type. However, it is hard to extend these approaches to branch-and-cut algorithm and to general MILPs (note that Rountree and Gillett [1982]'s method is applicable if the feasible region of each subsequent instance increases monotonically, that is, if the feasible region of next instance includes its predecessor's. Therefore, the cuts generated for the current instance will be valid for all subsequent instances and thus, they are not required to be modified). Note also that these algorithms need a priori information about each instance, i.e., the entire sequence is known head of time. This requirement makes it impossible to use them when the modifications of the input become available only after solving the initial instance. This is frequently the case in situations such as the solution of MILPs with parametric objective.

c. Analyze the final branch-and-bound tree of one problem to form a valid initial tree to invoke the branch-and-bound algorithm for another problem. The preceding discussion appeared in Roodman [October 1973] where only the pruned nodes of the previous tree are considered for a warm-start. Later, Nauss [1975] applied this approach to special parametric PILPs and MILPs with binary integer variables (knapsack, generalized assignment and capacitated facility location problems) and came up with promising computational results.

For branch-and-cut algorithm and for general MILPs, a warm-starting procedure can be derived as an extension of (c). Note that in the LP case, in order to start the simplex method to re-solve a new problem from a given basis, first feasibility and then optimality conditions are checked. For the MILP case, similarly, a subtree of the final branch-and-cut tree including the root node can be thought of as an initial basis for a modified problem and one needs to check if that can be translated into a valid starting partition to invoke branch-and-cut algorithm for the modified problem.

Below, we give a brief description of a warm-starting procedure and will discuss the implementation issues in Section 4.4.2. Assume that a branch-and-cut algorithm for an MILP is stopped either at optimality or after a limiting criteria is reached (such as a node limit, time limit, targeted duality gap etc.) and T^w, a subtree including the root node, is given as an input to re-solve a modified problem. Let T^w_L be the set of leaf nodes of T^w, $\tilde{\mathcal{S}}$ denote the feasible region of modified problem and $\tilde{\mathcal{S}}_t$ denote the feasible region of node $t \in T^w_L$ adapted to the problem modification. For instance, when the objective function is modified, the feasible regions are not affected. However when the right-hand side is changed from b to $\tilde{b}$, then $\tilde{\mathcal{S}} = \mathcal{S}(\tilde{b})$ and $\tilde{\mathcal{S}}_t = \mathcal{S}_t(\tilde{b})$ for each t.

1. *Feasibility test:* This is done in two steps:

   a. Check whether T^w is a *valid partition*, i.e., is $\tilde{\mathcal{S}} \subseteq \cup_{t \in T_L^W} \tilde{\mathcal{S}}_t$?

   b. Check the feasibility of optimal LP basis of each node $t \in T_L^w$ for the LP relaxation of modified node.

2. *Optimality test:* Check the sufficient optimality conditions described in Section 4.2 to see if a re-solve is needed.

3. *Re-Solve:* If (2) fails, modify each $t \in T_L^w$ to get a valid partition (apply the new changes and modify the cutting planes), apply primal or dual simplex pivoting to its previous basis to get the new optimal basis. Form a candidate list with these nodes and continue branch-and-cut algorithm.

Modification to the right-hand side. If $\tilde{b} \geq b$ and the initial feasible region is defined as $\mathcal{S}(b) = \{x \in \mathbb{Z}_+^r \times \mathbb{R}_+^{n-r} \mid Ax \geq b\}$, then (1a) can be skipped. Also in this case, the optimal LP bases of all tree nodes remain dual feasible to initiate (3). For other cases, due to generated cuts, the current partition may not be valid anymore. In particular, for a node $t \in T$, an inequality valid for $\mathcal{S}_t(b)$ might not be valid anymore for $\mathcal{S}_t(\tilde{b})$. As we have discussed in Section 2.2.5, if the subadditive representations or the right-hand side dependencies of the cuts are known, then the partition and the LP bases can be recovered by modifying these cuts to be valid for $\tilde{b}$. Otherwise, the right-hand side of each cut can be relaxed affinely by using proximity analysis until it is guaranteed not to cut off any feasible solution of the corresponding partition. For instance, possibly the worst case for this type of relaxation is to determine the right-hand side of each cut using the variable bounds as described in Theorem 2.21. Although this gives us a valid partition, makes each LP basis dual feasible, and enables warm-starting, cuts might become redundant and can be discarded. If neither can be done, one needs to check the validity of each cut. However, since validity checking is as difficult as generating new cuts, the easiest approach is to re-solve each modified node without cuts, get optimal LP bases and invoke the branch-and-cut algorithm.

Modification to the objective function. Clearly, any subtree together with cuts will form a valid partition. Hence (1a) will not be necessary. Furthermore, the optimal LP bases remain primal feasible and (3) can be invoked easily. On the other hand, sufficient conditions and sensitivity analysis can be checked over the whole tree T instead of over T^w since T is likely to yield better bounds.

Other modifications. Similarly, if the previous problem is a relaxation of the modified problem, then (1) can be skipped. Otherwise, one needs to check the validity of the partition, examining each node separately. Depending on the nature of the modification on the problem, appropriate actions such as updating/lifting the coefficients of cuts, changing branching decisions etc., might be needed at each node in order to obtain a valid partition. Although some cases such as adding columns or adding constraints or changing the variable bounds can be easily implemented, the problem structure can change completely and we might require one or more of those actions to be applied at the same time. In that case, the straightforward action is to trim the tree upwards beginning from the leaf nodes until a valid partition is obtained. Then, according to the modification, LP duality can be invoked at each leaf node of the trimmed tree to obtain primal or dual feasible bases to initiate warm-starting. We discuss some of these type of modifications and their implementations in Section 4.5).

4.4 Software and Implementation

We have implemented the techniques described in this chapter within the *SYMPHONY* software framework which is a customizable open-source MILP solver (see Ralphs and Güzelsoy [2005] for an overview and Mittelmann [2009a] for a comparison of noncommercial MILP solvers including SYMPHONY).

4.4.1 SYMPHONY

With its unique design, SYMPHONY was first introduced as a flexible, parallel solver for hard combinatorial problems. To facilitate parallel implementation and user customization, its functionality is divided into separate modules:

- the master module,
- the tree manager module,
- the cut generation module,
- the cut pool module,
- the node processing module,

The default solver behaviors can be altered with hundreds of parameters and over 50 *user callback* functions through which the user can have complete control over branching rules, cutting plane generation, management of the cut pool, processing of several tree nodes, diving strategies, and limited column generation. As for applications, SYMPHONY includes modules for solving the

- Traveling Salesman Problem,
- Vehicle Routing Problem,
- Set Partitioning Problem,
- Mixed Postman Problem,
- Matching Problem,
- Capacitated Network Routing Problem and
- Multi-Criteria Knapsack Problem.

(See Ralphs [2004] for further details and Ralphs et al. [2003b], Ralphs [2003], Ralphs et al. [2006] for published papers discussing SYMPHONY.)

Recently, SYMPHONY has been fully integrated with the libraries of the Computational Infrastructure for Operations Research (see COIN-OR [2009]) repository, including the Cut Generation Library (CGL) for generating cutting planes, and the Open Solver Interface (OSI), a C++ class that provides a standard API for accessing a variety of solvers. SYMPHONY is now implemented as a callable library and can be used as a stand-alone generic MILP solver accessible either through calls to the native C subroutines of the API or through a C++ class derived from the COIN-OR OSI.

Below, we briefly describe the callable library, main subroutines and the C++ interface to access these subroutines (see Ralphs and Güzelsoy [2005] for more details).

Callable library. SYMPHONY's callable library consists of a complete set of subroutines for loading and modifying problem data, setting parameters, and invoking solution algorithms. The user invokes these subroutines through the API specified in the header file **symphony_api.h**. Some of the basic commands are described below.

sym_open_environment(): Opens a new environment, consisting of a new instance of the master module, and returns a pointer to it. This pointer then has to be passed as an argument to all other API subroutines.

sym_parse_command_line(): Invokes the built-in parser for setting commonly used parameters, such as the file name which to read the problem data, via command-line switches. A call to this subroutine instructs SYMPHONY to parse the command line and set the appropriate parameters. This subroutine also sets all other parameter values to their defaults, so it should only called when this is desired.

sym_load_problem(): Reads the problem data and sets up the root subproblem. This includes specifying the cuts and variables in the *core* (those that are initially present in every subproblem during the search process) and the additional cuts and variables to be initially active in the root subproblem.

```
int main(int argc, char **argv)
{
   sym_environment *p = sym_open_environment();
   sym_parse_command_line(p, argc, argv);
   sym_load_problem(p);
   sym_solve(p);
   sym_close_environment(p);
}
```

Figure 4.1: Implementation of a generic MILP solver with the SYMPHONY C callable library.

sym_explicit_load_problem(): By default, SYMPHONY reads an MPS or GMPL/AMPL file specified by the user, but the user can override this default by defining the problem explicitly with this function or implementing a user callback that reads the data from a file in a customized format.

sym_solve(): Solves the currently loaded problem from scratch.

sym_resolve(): Solves the currently loaded problem from a warm start (see Section 4.4.2).

sym_mc_solve(): Solves the currently loaded problem as a bicriteria problem (see Section 4.5.6.

sym_close_environment(): Frees all problem data and deletes the master module.

As an example of the use of the library functions, Figure 4.1 shows the code for implementing a generic MILP solver with default parameter settings. To read in an MPS file called *model.mps* and solve it using this program, the following command would be issued:

```
symphony -F model.mps
```

The user does not have to invoke a command to read the MPS file. During the call to **sym_parse_command_line()**, SYMPHONY learns that the user wants to read in an MPS file. During the subsequent call to **sym_load_problem()**, the file is read and the problem data stored. To read an GMPL/AMPL file, the user would issue the command

```
symphony -F model.mod -D model.dat
```

Although the same command-line switch is used to specify the model file, the additional presence of the **-D** option indicates to SYMPHONY that the model file is in GMPL/AMPL format and the GMPL parser is invoked (Makhorin [2009]). Note that the interface and the code of Figure 4.1 is the same for both sequential and parallel computations. The choice between sequential and parallel execution modes is made at compile-time through modification of the makefile or the project settings, depending on the operating system (see Ralphs [2006] for using SYMPHONY as a generic MILP solver in parallel environment).

OSI interface. The Open Solver Interface (OSI) is a C++ class that provides a standard API for accessing a variety of solvers for mathematical programs. It is provided as part of the COIN-OR repository, along with a collection of solver-specific derived classes that translate OSI call into calls to the underlying solver's library. A code implemented using calls to the methods in the OSI base class can easily be linked with any solver for which there is an OSI interface. This allows development of solver-independent codes and eliminates many portability issues. The current incarnation of OSI supports only solvers for linear and mixed-integer linear programs, although a new version supporting a wider variety of solvers is currently under development.

We have implemented an OSI interface for SYMPHONY that allows any solver built with SYMPHONY to be accessed through the OSI, including customized solvers and those configured to run on parallel architectures. To ease code maintenance, for each method in the OSI base class, there is a corresponding method in the callable library. The OSI methods are implemented simply as wrapped calls to the SYMPHONY callable library. When an instance of the OSI interface class is constructed, a call is made to **sym_open_environment()** and a pointer to the master module is stored in the class. Most subsequent calls within the class can then be made without any arguments. When the OSI object is destroyed, **sym_close_environment()** is called and the master module is destroyed.

To fully support SYMPHONY's capabilities, we have extended the OSI interface to include some methods not in the base class. For example, we added calls equivalent to our **sym_parse_**

```
int main(int argc, char **argv)
{
   OsiSymSolverInterface si;
   si.parseCommandLine(argc, argv);
   si.loadProblem();
   si.initialSolve();
}
```

Figure 4.2: Implementation of a generic MILP solver with the SYMPHONY OSI interface.

command_line(). Figure 4.2 shows the program of Figure 4.1 implemented using the OSI interface.

4.4.2 Implementation

We have added to SYMPHONY warm-starting capability and basic sensitivity analysis tools. For now, because of the complications we have described before, these methods can be invoked only for rim vectors modifications for generic problems. However, we have also enabled warm-starting for different parameter changes (column generation and variable bound changes) for specific applications as we will describe in Section 4.5).

Warm Starting. SYMPHONY utilizes a branch-and-cut approach to solve an MILP problem. A priority queue of candidate subproblems, the leaf nodes of a continuously growing branch-and-bound tree, is available to be processed and the algorithm terminates when this queue is empty or when another pre-defined condition is satisfied. When the user asks SYMPHONY to keep information to be used for warm-starting, the description of the tree as well as the other auxiliary data needed to restart the computation is stored in the warm-start structure we have implemented. This description contains complete information about the subproblem corresponding to each node in the search tree, including the branching decisions that lead to the creation of the node, the list of active variables and constraints, and warm-start information for the subproblem itself (which is a linear program). All information is stored compactly using SYMPHONY's native data structures by storing only the differences between a child and its parent rather than an explicit description of every node. This approach reduces the tree's description to a fraction of the size it would otherwise be. In addition to the tree itself, other relevant information regarding the status of the computation is recorded, such as the current bounds and best feasible solution found so far. Using the warm-start class, the user can save a warm-start to disk, read one from disk, or restart the computation from any warm-start after modifying parameters or the problem data itself. This allows the user to easily implement periodic check pointing, to design dynamic algorithms in which the parameters are modified after the gap reaches a certain threshold, or to modify problem data during the solution process if needed.

Modifying Parameters. The most straightforward use of the warm-start class is to restart the solver after modifying problem parameters. The solution process can be interrupted with a satisfied condition and some regulators of the algorithm are changed. For instance, after a predefined time or node limit or a targeted duality gap is reached, according to the current progress one may want to change some of branching or node selection rules. In this case, the master module automatically records the warm-start information resulting from the last solve call and restarts from that checkpoint if a call to **resolve()** is made, unless external warm-start information is loaded manually. To start the computation from a given warm-start when the problem data has not been modified, the tree manager simply traverses the tree and loads in leaf nodes marked as candidates for processing. Once the node queue has been reformed, the algorithm picks up where it left off. Figure 4.3 illustrates this concept by showing the code for implementing a solver that changes from depth first search to best first search after the first feasible solution is found. The situation is more challenging if the user modifies problem data in between calls to the solver. We address this situation next.

Modifying Problem Data. If the user modifies problem data in between calls to the solver, SYMPHONY must make corresponding modifications to the leaves of the current search tree to allow execution of the algorithm to continue. To initialize the algorithm, we first decide which subtree of the final tree should be used (see Section 4.3). For now, the user can choose one of these options:

```
int main(int argc, char **argv)
{
   OsiSymSolverInterface si;
   si.parseCommandLine(argc, argv);
   si.loadProblem();
   si.setSymParam(OsiSymFindFirstFeasible, true);
   si.setSymParam(OsiSymSearchStrategy, DEPTH_FIRST_SEARCH);
   si.setSymParam(OsiSymKeepWarmStart, true);
   si.initialSolve();
   si.setSymParam(OsiSymFindFirstFeasible, false);
   si.setSymParam(OsiSymSearchStrategy, BEST_FIRST_SEARCH);
   si.resolve();
}
```

Figure 4.3: Implementation of a dynamic MILP solver with SYMPHONY.

- Take the first k nodes.
- Take all nodes above level k in the tree.
- Take the first $p\%$ of the nodes.
- Take all nodes above the level $p\%$ of the tree depth.
- The subtree that includes the paths yielding feasible solutions.
- The subtree that includes the paths along which a given subset of variables were branched on.

If any of these are chosen, all nodes but those in the chosen subtree are discarded. Then, we make the appropriate changes to the subproblem of each leaf node to ensure the validity of this subtree for the new problem. For instance, for the changes on right-hand side, we need to validate for a leaf node t the cuts added so far along the path from root node until this node. To give an idea, consider the Gomory mixed integer cuts obtained by the procedure (Nemhauser and Wolsey [1988])

$$\sum_{j \in I} F_\alpha(\lambda A_j) + \sum_{j \in C} \bar{F}_\alpha(\lambda A_j) \geq F_\alpha(\lambda b), \tag{4.15}$$

where $\lambda \in \mathbb{R}^m_+$, $0 \leq \alpha < 1$, and F_α is the corresponding subadditive function defined as

$$F_\alpha(h) = \begin{cases} h - \lfloor h \rfloor & h - \lfloor h \rfloor \leq \alpha \\ h - \left(\lfloor h \rfloor + \frac{\max\{h - \lfloor h \rfloor, 0\} - \alpha}{1-\alpha} \right) & h - \lfloor h \rfloor > \alpha \ , \end{cases}$$

with its upper directional derivative

$$\bar{F}_\alpha(h) = \begin{cases} h & h \geq 0 \\ \frac{h}{1-\alpha} & h < 0 \ . \end{cases}$$

In implementation, we apply (4.15) to a row of the simplex tableau of

$$x_B + (A_B)^{-1} A_{N \backslash B} x_{N \backslash B} = (A_B)^{-1} b \ \ , x \in \mathbb{Z}^r_+ \times \mathbb{R}^{n-r}_+,$$

where B is the set of indices in the current basis. Assume that Gomory mixed integer cut is extracted from row i of this system for a fixed $0 \leq \beta < 1$ by F_β. Then, when b is changed to $\tilde{b}$, the inequality (4.15) remains valid as long as its right-hand side is changed to $F_\beta((A_B)^{-1}_i \tilde{b})$ where $(A_B)^{-1}_i$ is the i^{th} row of $(A_B)^{-1}$. Note that although we have illustrated above the validation of a Gomory mixed integer cut derived from the simplex tableau of the original constraint set $\{Ax = b\}$, one also need to keep track of the recursive dependency of each cut to the previously generated cuts as described in Section 2.2.5.

After the modifications, the computation is warm started from this given subtree. Each leaf node, regardless of its status after termination of the previous solve call, must be inserted into the queue of candidate nodes and reprocessed with the changed rim vectors. After this reprocessing, the computation can continue as usual. Figure 4.4 shows the necessary code for warm-starting with the nodes above the 6th level of the final tree of the previous computation. On the other hand, in Figure 4.5, the solver is allowed to process 100 nodes before the warm-start information is saved. Afterwards, the original problem is modified and re-solved from this saved checkpoint.

```
int main(int argc, char **argv)
{
   OsiSymSolverInterface si;
   si.parseCommandLine(argc, argv);
   si.setSymParam(OsiSymKeepWarmStart, true);
   si.initialSolve();
   for(int i=100; i<110; i++){
     si.setObjCoeff(i, 400);
   }
   si.setSymParam(OsiSymWarmStartTreeLevel, 6);
   si.resolve();
}
```

Figure 4.4: Use of SYMPHONY's warm-starting capability.

```
int main(int argc, char **argv)
{
   OsiSymSolverInterface si;
   CoinWarmStart * ws;
   si.parseCommandLine(argc, argv);
   si.setSymParam(OsiSymNodeLimit, 100);
   si.setSymParam(OsiSymKeepWarmStart, true);
   si.initialSolve();
   ws = si.getWarmStart();
   si.setSymParam(OsiSymNodeLimit, -1);
   si.resolve();
   for(int i=100; i<110; i++){
     si.setObjCoeff(i, 400);
   }
   si.setWarmStart(ws);
   si.resolve();
   si.initialSolve();
}
```

Figure 4.5: Use of SYMPHONY's warm-starting capability.

Sensitivity Analysis. Our warm-start structure is also equipped to collect more information, namely, the feasible solutions found so far and the dual solutions of each node. In order to save memory, these solutions are kept in sparse form (nonzero elements and their indices). For any modification, upper and lower bounds are obtained as described in Section 4.2. For modifications to the right-hand-side, the lower bound is obtained using Theorem 4.2. Figure 4.6 shows an example of a program that uses this sensitivity analysis function. This code will give a lower bound obtained by (4.6) for a modified problem with new right hand side values of *7000* and *6000* in the fourth and seventh rows.

```
int main(int argc, char **argv)
{
   OsiSymSolverInterface si;
   si.parseCommandLine(argc, argv);
   si.loadProblem();
   si.setSymParam(OsiSymSensitivityAnalysis, true);
   si.initialSolve;
   int ind[2];
   double val[2];
   ind[0] = 4;   val[0] = 7000;
   ind[1] = 7;   val[1] = 6000;
   double lb = si.getLbForNewRhs(ind, ind, val);
}
```

Figure 4.6: Performing sensitivity analysis with SYMPHONY

4.5 Experimental Results

As a small demonstration of warm-starting, we tested the code of Figure 4.5 with the file **p0201** (Bixby et al. [1992]), where the original coefficients vary between 300 and 500. The results are presented in Table 4.1. As for the generated tree nodes, observe that the warm-starting procedure analyzes more of them than would have been required starting from scratch. Nevertheless, solving the problem from a warm-start decreases the solution time significantly.

	CPU Time	Search Tree Nodes
Generate warm-start	16	100
Solve modified problem (from scratch)	27	166
Solve modified problem (from warm-start)	4	195

Table 4.1: Warm starting a computation with **p0201**

4.5.1 General MILP

We have done a variety of tests with a subset of the well-known MIPLIB3 (Bixby et al. [1998]) test set to understand how well warm-starting works for general MILPs. In each of the following graphs, the horizontal axis represents the different instances whereas the vertical axis is the solution times. Figures 4.7, 4.8 and 4.9 are the results of different options for warm starting of each instance after perturbing a random subset of objective coefficients of random size. In Figure 4.7, for instance, $r = 0$ means that the instance is solved from scratch, $r = 50$ means it is solved from the subtree consisting of nodes whose depth is half the original tree depth and $r = 100$ means it is solved from the complete tree of the original problem. On the other hand, Figures 4.10 and 4.11 consist of the results of warm-starting with option $r = 100$ after modifying a random subset of objective coefficients with fixed size.

Figures from 4.12 through 4.15 are the results of warm-starting after perturbing the right-hand side. In Figure 4.12, the test instance is a single knapsack problem and the right-hand side is changed between $b/2$ and $3b/2$ and for each right-hand side, warm-starting is initiated with option $r = 25$. In the rest of the experiment, each figure consists the results of warm-starting with Gomory mixed integer cuts (see 4.15) after modifying a random subset of right-hand sides of a set of test instances. For each instance, warm-starting is initiated with option $r = 100$. One difficulty with randomly changing the right-hand side is that the experiment may fail to capture the true potential of warm-starting because the modified problem may be too easy to solve compared to the original problem. Such a case might appear if the LP relaxation of the modified problem, and hence the the modified problem itself, is infeasible. Note that our warm-starting procedure would be slower in this case, since it requires first to walk through all the leaf nodes whereas solving from scratch only requires to solve the LP relaxation of the root node. In order to prevent that, we change the right-hand side in a way that the modified problem remains feasible and is as hard to solve as the original problem.

For small modifications, warm-starting seems to yield good results. However, its efficiency tends to decrease as the size of the modification increases. In addition, observe the behavior of warm-starting with different options. Although the option of warm-starting with the complete tree seems to be dominating, this is not true in some cases. For the knapsack instance, $r = 25$ seemed to be the best among all options.

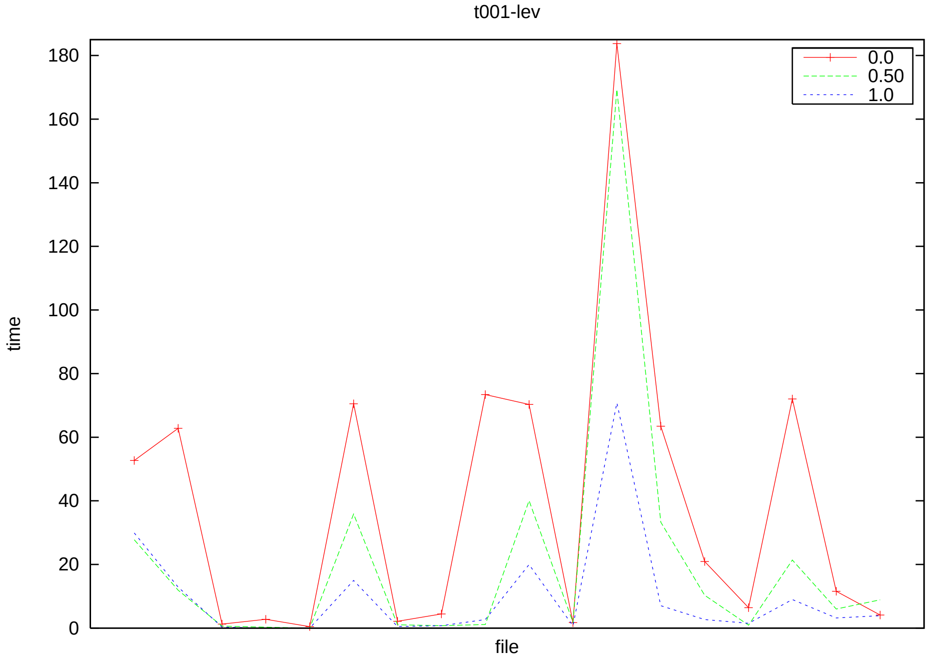


Figure 4.7: Warm start after 1% modification on a random subset of objective coefficients of random size. Warm-start tree consists of nodes above the $r\%$ level of the tree, $r \in \{0, 50, 100\}$.

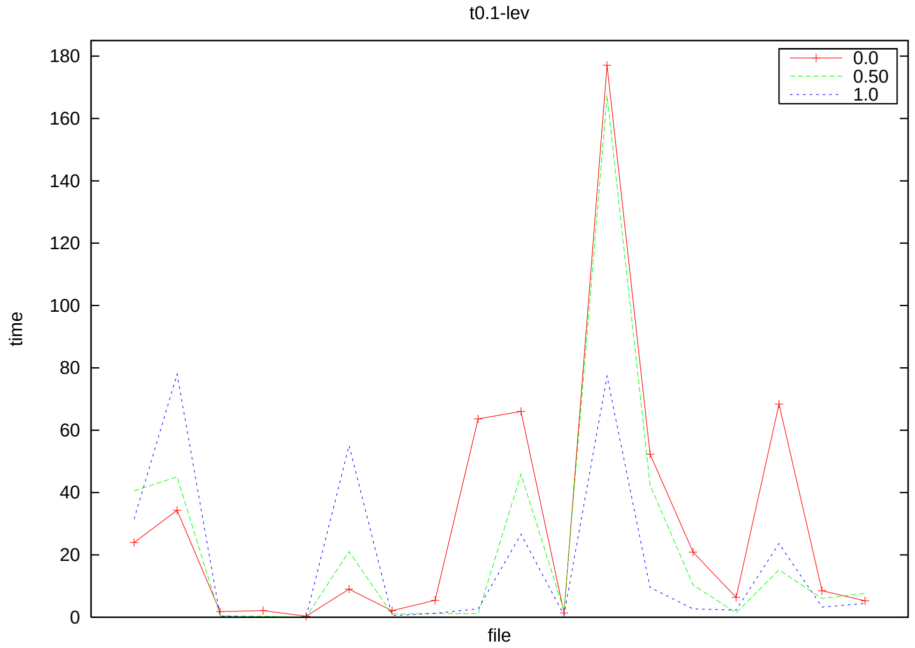


Figure 4.8: Warm start after 10% modification on a random subset of objective coefficients of random size. Warm-start tree consists of nodes above the $r\%$ level of the tree, $r \in \{0, 50, 100\}$.

The ability to re-solve after modifying problem data is not only useful for sensitivity analysis but also has a wide range of applications in practice where a sequence of related MILP problems must be solved. For instance, most of the algorithms used to analyze

- decomposition methods,
- column - row generation methods,
- stochastic MILP problems,
- parametric MILP problems,
- multicriteria MILP problems, and
- feasibility problems or determining irreducible infeasible subsystems

depend on solving a family of integer programs. To demonstrate the effect of the warm-starting capability, we have implemented algorithms for solving iterative combinatorial auctions, capacitated

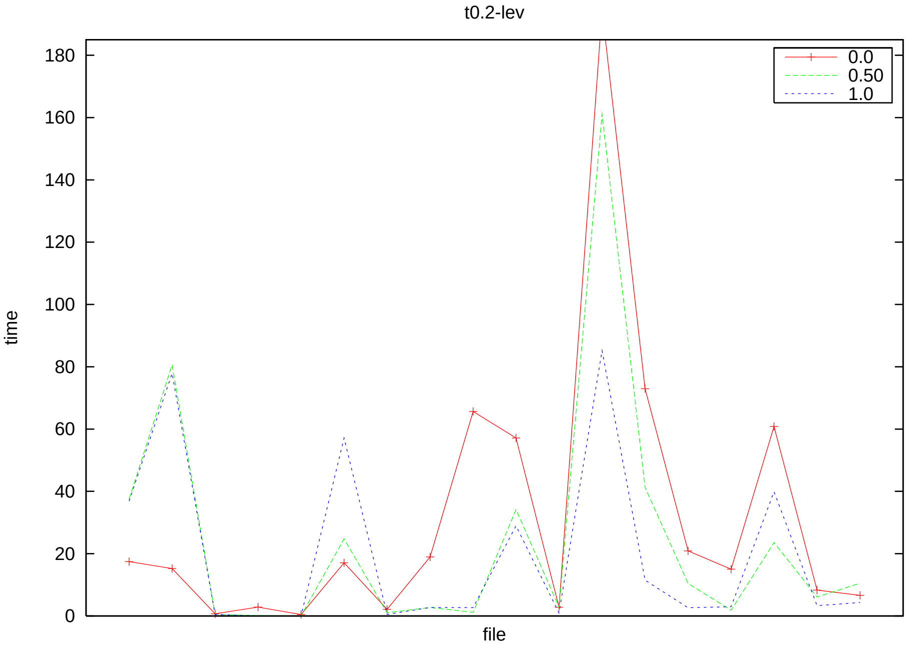
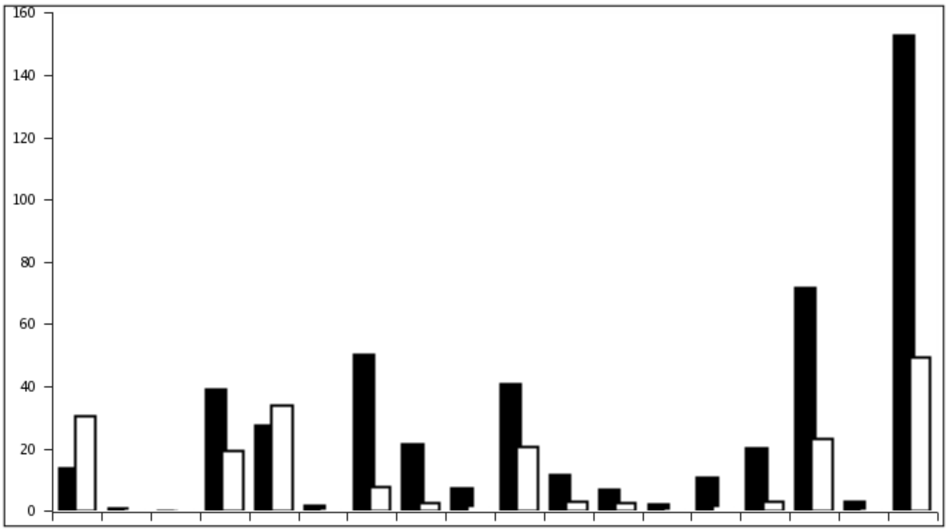
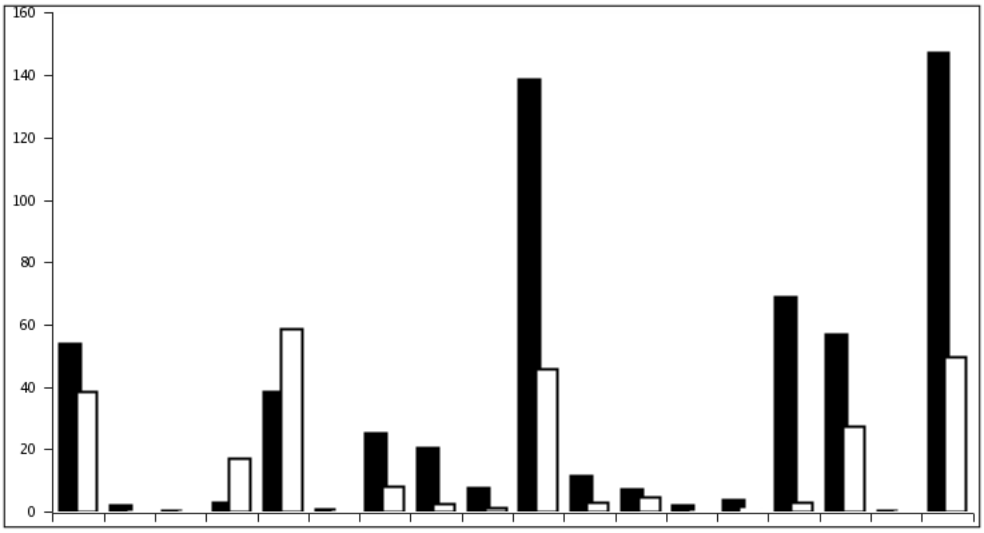


Figure 4.9: Warm start after 20% modification on a random subset of objective coefficients of random size and use the nodes above the $r\%$ level of the tree, $r \in \{0, 50, 100\}$.



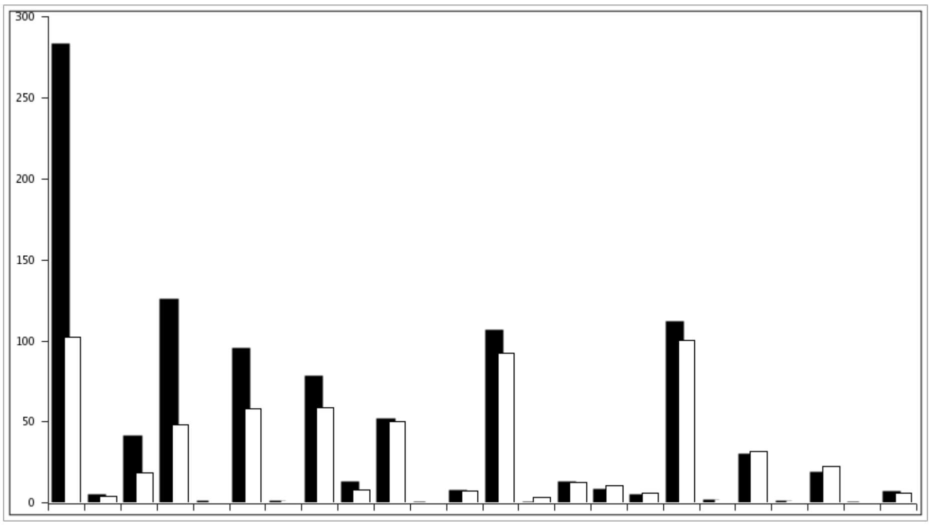
Black: without warm-starting
White: with warm-starting

Figure 4.10: Warm start after random perturbation of $+/-10\%$ on a random subset of objective coefficients of size $0.1n$.



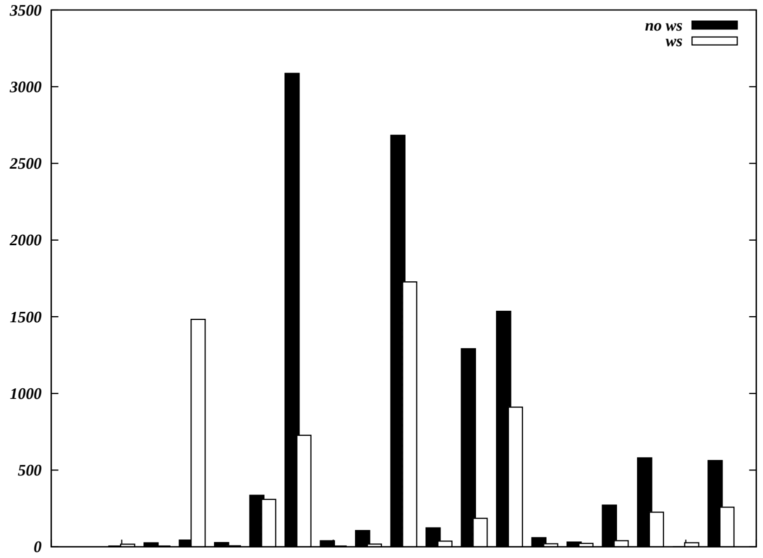
Black: without warm-starting
White: with warm-starting

Figure 4.11: Warm start after random perturbation of $+/-10\%$ on a random subset of objective coefficients of size $0.2n$.



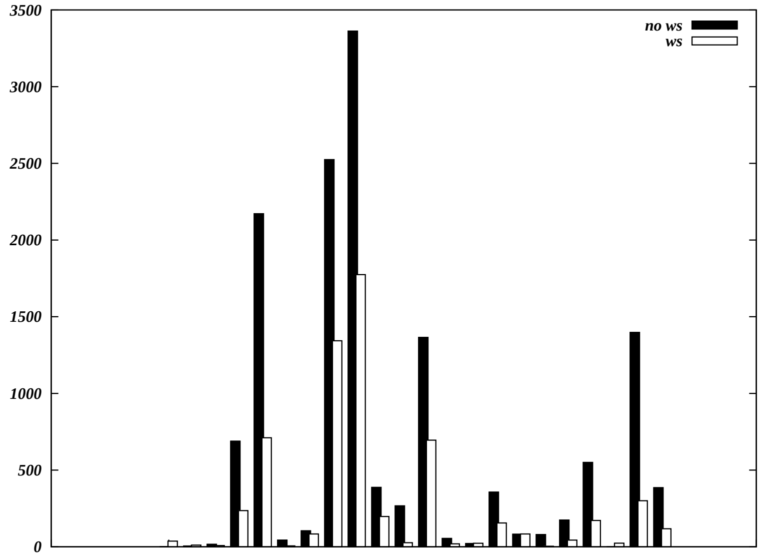
Black: without warm-starting
White: with warm-starting

Figure 4.12: Change right-hand side b of a knapsack problem between $b/2$ and $3b/2$ and warm start using the nodes above the 25% level of the tree.



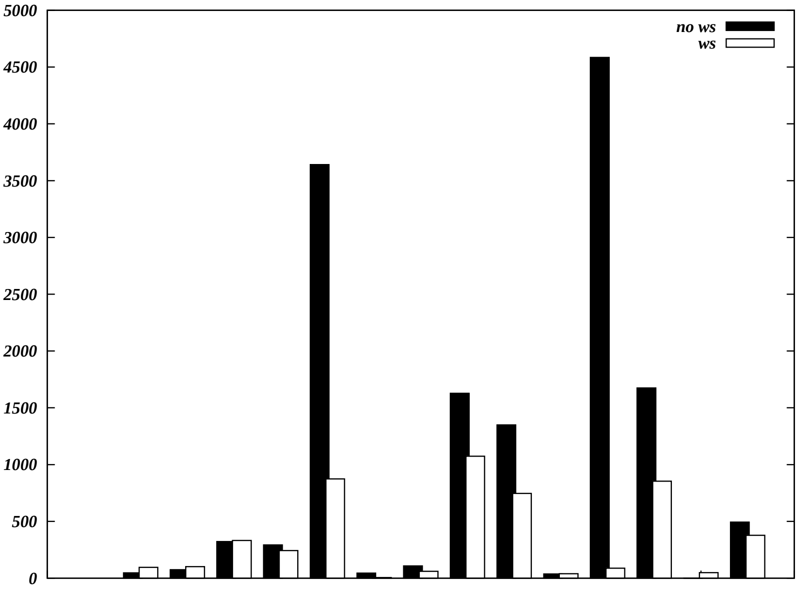
Black: without warm-starting
White: with warm-starting

Figure 4.13: Warm start after random perturbation of $+/-10\%$ on a random subset of right-hand side of size $0.01m$.



Black: without warm-starting
White: with warm-starting

Figure 4.14: Warm start after random perturbation of $+/-10\%$ on a random subset of right-hand side of size $0.05m$.



Black: without warm-starting
White: with warm-starting

Figure 4.15: Warm start after random perturbation of $+/-10\%$ on a random subset of right-hand side of size $0.1m$.

vehicle routing problems, feasibility algorithms (RINS), two-stage stochastic integer and bicriteria integer programs. Next, we briefly discuss these algorithms and then present our computational results.

4.5.2 Iterative Combinatorial Auctions

In combinatorial auctions, we iteratively try to match a set of packages of items with bidders in a way that the revenue for the auctioneer is maximized. Let S be the set of items being auctioned. Then at each round t, we need to solve the following *winning determination problem* (WDP):

$$\begin{aligned} \max \quad & \sum_{j \in S^t} c_j x_j \\ s.t. \quad & \sum_{j \in S^t} a_{ij} x_j \leq 1, \quad \forall i \in S \\ & x_j \in \{0,1\}, \quad \forall j \in S^t, \end{aligned} \tag{4.16}$$

where S^t is the set of considered packages in round t, c_j is the bid amount of package j,

$$a_{ij} = \begin{cases} 1, & \text{if item } i \text{ is in package } j; \\ 0, & \text{otherwise.} \end{cases},$$

and

$$x_j = \begin{cases} 1, & \text{if package } j \text{ is a winning package;} \\ 0, & \text{otherwise.} \end{cases}.$$

In the above set packing formulation, bidders who have submitted bids on the winning packages are awarded the corresponding items in those packages. Constraints ensure that each item is awarded at most once. Depending on our assumptions, we may also formulate WDP as a set partitioning problem

$$\begin{aligned} \max \quad & \sum_{j \in S^t} c_j x_j \\ s.t. \quad & \sum_{j \in S^t} a_{ij} x_j = 1 \quad \forall i \in S \\ & x_j \in \{0, 1\} \quad \forall j \in S^t \end{aligned} \tag{4.17}$$

so that all items are ensured to be awarded in round t. Although this formulation is used under the assumption of zero disposal value, bidders are also allowed to win more than one package in each round.

At each round, auctioneer solves the WDP and provides bidders with feedback information. Receiving this feedback after each round, the non-winning bidders can

- increase their bids for the packages already in the auction, or
- create bids for new packages.

according to the auction rules Therefore, in the next round, we need to solve a modified WDP with updated objective coefficients, new columns or both.

In our computational experiments, we use SYMPHONY's SPP+CUTS package which generates at each node star cliques, odd holes and odd anti-holes cuts (Eso [1999]) to solve set partitioning or packing problem. These cuts remain valid in our case for the branching tree from round-to-round since they are generated from the constraint matrix and we only change the objective coefficients or add new columns. We do lift the star cliques inequalities greedily. That is when new columns are added, we check whether the inequalities can be lifted with the new variables representing added columns.

We test warm-starting for both set packing and set partitioning cases for auction instances generated by Combinatorial Auction Test Suite (CATS - Brown et al. [2007]) including number of bidders varying between 5 and 10 and number of items varying between 3 and 18. Note that the smallest instance at any round might have at most $5 * (2^3 - 1)$ columns whereas the largest instance might have at most $10 * (2^{18} - 1)$ columns.

In each case, we try resetting the warm-start tree either at each round (Figure 4.16, 4.18) or dynamically (Figures 4.17, 4.19). In the latter case, though many other rules specific to the structure of the auction problem can be used, we only reset the warm-start at the end of current round if the total # of columns added since the last reset is over $p*$(# of bidders) or the solution time for the current round is over $q*$(the solution time) of the WDP solved in the very next round of the last reset. For the following tests, the best outcome we have obtained was for $p = 5$ and $q = 1.25$ and as a result, we have observed the reset counts for Figure 4.17:

WS-20% → 8.8% (59/672)

WS-30% → 8.7% (58/668)

WS-50% → 13.2% (36/272)

WS-100% → 9.2% (25/272),

and for Figure 4.19:

WS-20% → 9.3% (27/290)

WS-30% → 8.9% (44/497)

WS-50% → 11.4% (33/290)

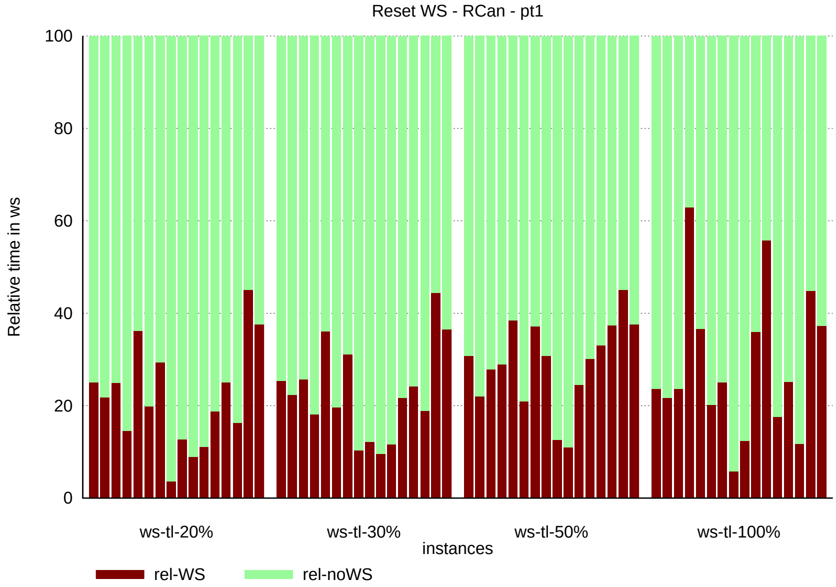


Figure 4.16: WDP-Set Packing Formulation / Reset WS each round / WS consists of nodes below the $r\%$ level of the tree, $r \in \{20, 30, 50, 100\}$

$$\text{WS-100\%} \rightarrow 8.9\% \text{ (26/289)},$$

where WS-r denotes the corresponding case where only the nodes below level $r\%$ of the branching tree used as a warm-start. Each column consists of the normalized solution times for the corresponding instance. In other words, if we let for a specific instance λ_{WS} be the total solution time with warm-starting, λ_{NoWS} be the total solution time without warm-starting, then *rel-WS* = $100 * \lambda_{WS}/(\lambda_{WS} + \lambda_{NoWS}$ and *rel-noWS* = 100 - rel-WS.

In all experiments, warm-starting for both strategies is dominating solving the instances from scratch. For set packing formulation, warm-starting at each round with a small tree size seems to be the best option. For set partitioning formulation, although both strategies are close to each other, the best option seems to warm start dynamically with a large tree size. One reason for this result is that the feasible solution space is restricted with equality constraints and the number of partitions of the feasible region in a larger tree helps to identify them quicker.

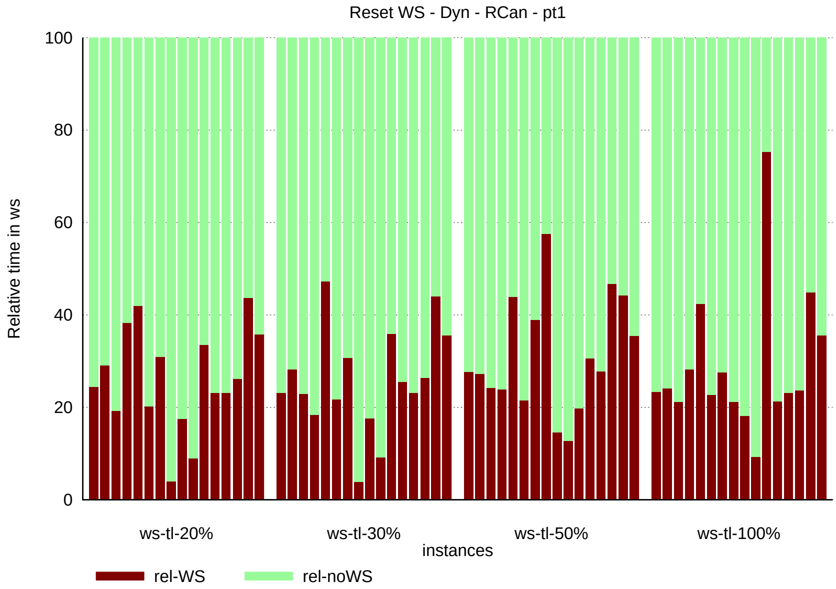


Figure 4.17: WDP-Set Packing Formulation / Reset WS dynamically / WS consists of nodes below the $r\%$ level of the tree, $r \in \{20, 30, 50, 100\}$

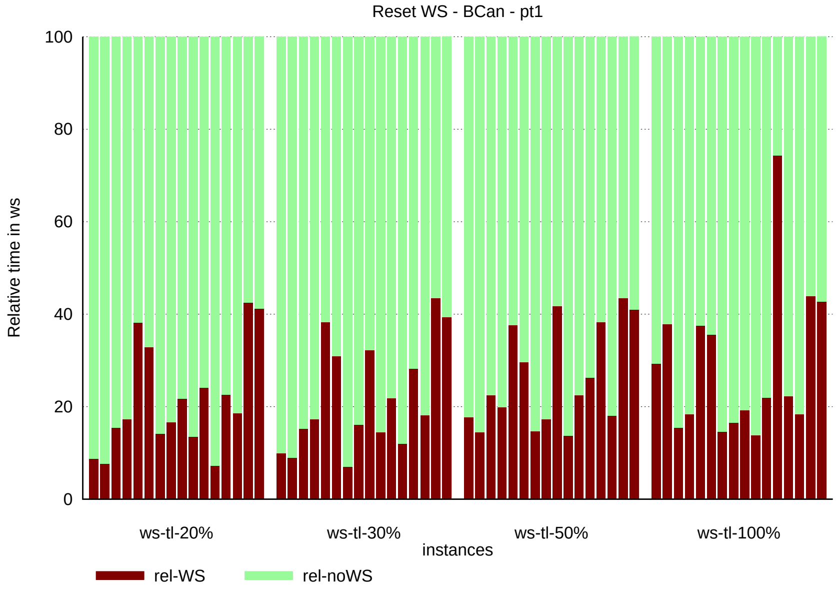


Figure 4.18: WDP-Set Partitioning Formulation / Reset WS each round / WS consists of nodes below the $r\%$ level of the tree, $r \in \{20, 30, 50, 100\}$

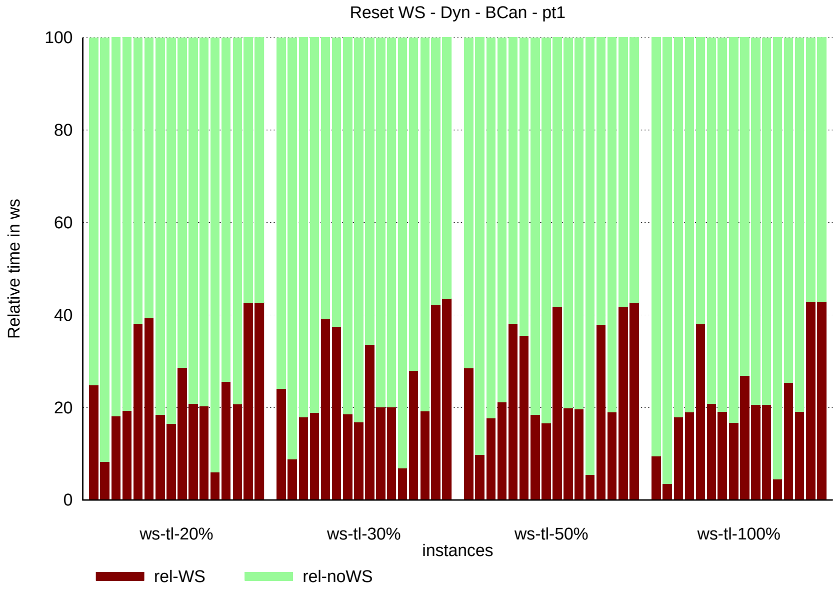


Figure 4.19: WDP-Set Partitioning Formulation / Reset WS dynamically / WS consists of nodes below the $r\%$ level of the tree, $r \in \{20, 30, 50, 100\}$

4.5.3 Capacitated Vehicle Routing Problems

In the *capacitated vehicle routing problem* (VRP), there is a set N of customers and a quantity d_i of a single commodity is to be delivered from a central depot indexed at 0 to each customer $i \in N = \{1, \dots, n\}$ with k independent delivery vehicles of identical capacity C. The delivery is to be accomplished at minimum total cost, with $c_{ij} \geq 0$ denoting the transit cost from i to j, $i, j \in N \cup 0$. The cost structure is assumed to be symmetric, that is, $c_{ij} = c_{ji}$, $c_{ii} = 0$ and a feasible solution is a partition of N into k routes $\{R_1, \dots, R_k\}$, each satisfying $\sum_{j \in R_i} d_j \leq C$.

One approach to formulate VRP is to consider a complete undirected graph with nodes $V \equiv N \cup \{0\}$, edges E, edge-traversal costs c_{ij}, $\{i, j\} \in V$ where a feasible solution is the union of k cycles whose intersection is the depot node. The corresponding IP formulation is

$$
\begin{array}{rcl}
\min \sum_{e \in E} c_e x_e & & \\
\sum_{e=\{0,j\} \in E} x_e & = & 2k \\
\sum_{e=\{i,j\} \in E} x_e & = & 2 \;\; \forall i \in N \\
\sum_{\substack{e=\{i,j\} \in E, \\ i \in S, j \notin S}} x_e & \geq & 2b(S) \;\; \forall S \subset N, |S| > 1 \\
0 \leq x_e & \leq & 1 \;\; \forall e = \{i,j\} \in E, i,j \neq 0 \\
0 \leq x_e & \leq & 2 \;\; \forall e = \{0,j\} \in E \\
x_e & & \text{integral} \;\; \forall e \in E
\end{array}
$$

where the first two sets of constraints are the degree constraints, $b(S)$ is a lower bound on the number of trucks needed to service the customers in set S and hence, the third set is the capacity constraints to ensure that the solution is connected and that no route has total demand exceeding the capacity C.

Considering the number of possible capacity constraints, it is clear that the above formulation itself is intractable. In fact, even solving the complete LP relaxation is proved to be an NP-complete problem. Ralphs et al. [2003a] outlines a branch-and-cut algorithm starting with the relaxed formulation including only the degree and bounding constraints. At each node before branching, infeasible LP relaxation solutions are separated from the current polytope by cuts derived from fractional graphs. In this approach, the generated cuts are capacity constraints with $b(S) = \lceil (\sum_{i \in S} d_i)/C \rceil$.

In real-time environment, there might be a need for re-solving a modified VRP instance due to some of the following scenarios:

- number of available trucks might change,

- new customers might show up,

- customers might cancel/change their orders,

Though it is possible to use warm-starting for any of the scenarios above, we only consider the last one and analyze the use of warm-starting for that case.

Note that the warm-start information remains valid as long as the customer demands remain the same. Otherwise, we first need to validate the right hand side of each valid inequality derived from the set $S \subset N, |S| > 1$ for the new demand $\bar{d}_i, i \in S$ by replacing their right-hand sides with

$$\bar{b}(S) = \left\lceil (\sum_{i \in S} \bar{d}_i)/C \right\rceil . \tag{4.18}$$

In addition, if customer l cancels their demand, we

- either set $\bar{d}_l = 0$ and update the degree constraint $\sum_{e=\{l,j\} \in E} x_e = 0$,
- or set $\bar{d}_l = C$, introduce a dummy truck and update the degree constraint $\sum_{e=\{0,j\} \in E} x_e = 2(k+1)$,

so that the customer l is excluded from the search space, capacity constraints and dual bases of tree nodes remain valid for warm-starting.

In our experiments, we use SYMPHONY's VRP package and a subset of VRPLIB (VRPLIB [2006]) instances with number of trucks $k \in \{3, \ldots, 7\}$ and number of customers $|N|$ varying between 30 and 57. For each instance, we test our warm-starting for different scenarios. For each scenario, the setup consists of randomly selecting $\alpha = 5\%$ or $\alpha = 10\%$ of customers and perturbing their demands within $\beta = \pm 5\%$ (Figures 4.20, 4.23), $\beta = \pm 10\%$ (Figures 4.21, 4.24), or $\beta = \pm 20\%$ (Figures 4.22, 4.25). For an instance and an experiment setup, the reported time is the total time for solving at most ten different scenarios within a time limit of 7200 seconds.

For the setups with small modifications on customer demands, i.e., $\alpha = \pm 5\%$ with $\beta \in \{\pm 5, \pm 10\}\%$ and $\alpha = \pm 10\%$ with $\beta = \pm 5\%$, warm-starting yields better running times for almost all instances, with the improvements of over a factor of 5 for some cases. However, as deviation in customer demands increases, warm-starting becomes slower for obvious reasons. Notice that this result can be observed clearly in the last setup for which we have the largest deviation of $\alpha = \pm 10\%$ with $\beta = \pm 20$.

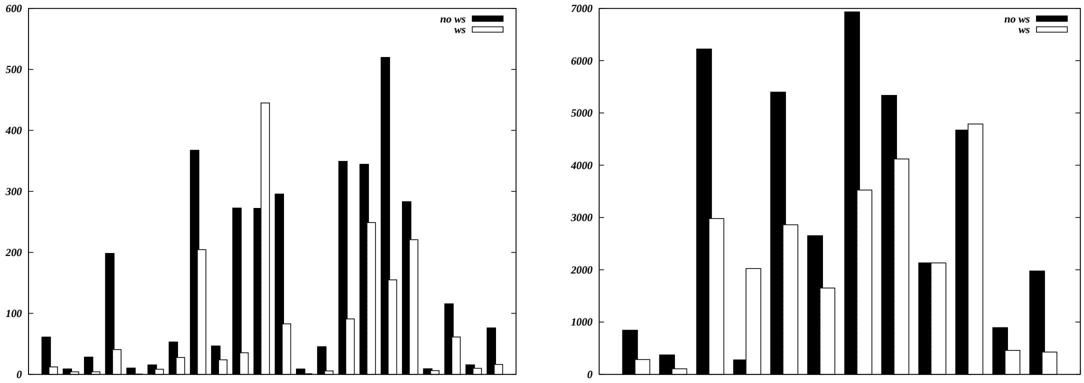


Figure 4.20: Warm-starting for VRP: $\alpha = 5\%, \beta = \pm 5\%$.

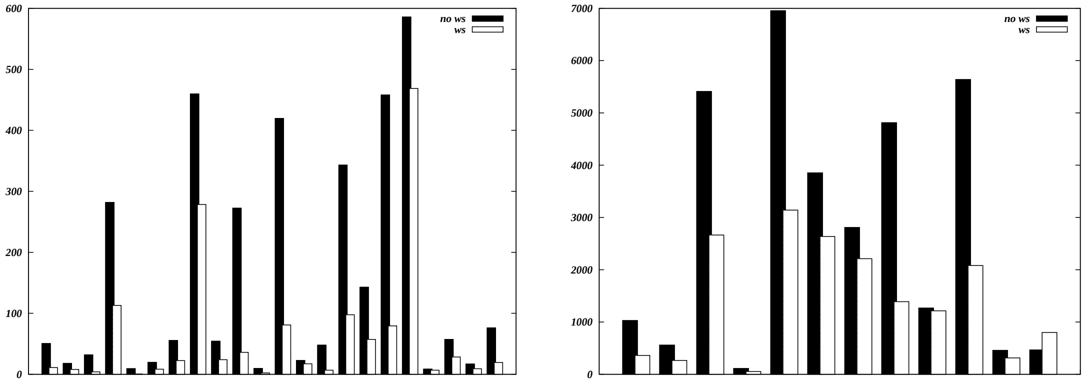


Figure 4.21: Warm-starting for VRP: $\alpha = 5\%, \beta = \pm 10\%$.

4.5.4 Primal Heuristics - RINS

Relaxation Induced Neighborhood Search (RINS) is an improvement heuristic for MILPs proposed by Danna et al. [2005]. It is based on exploring a *search neighborhood* based on the structure of an incumbent solution and the LP solution of a given node in the branch-and-cut search tree. In particular, suppose for a given MILP that x^* is a feasible solution and $\bar{x}$ is obtained after solving the LP relaxation at one of the nodes of the branching tree. Then, a restricted MILP formed by fixing the variables x_i $i \in I$ for which $\bar{x}_i = x_i^*$ is solved in the hope of finding a better

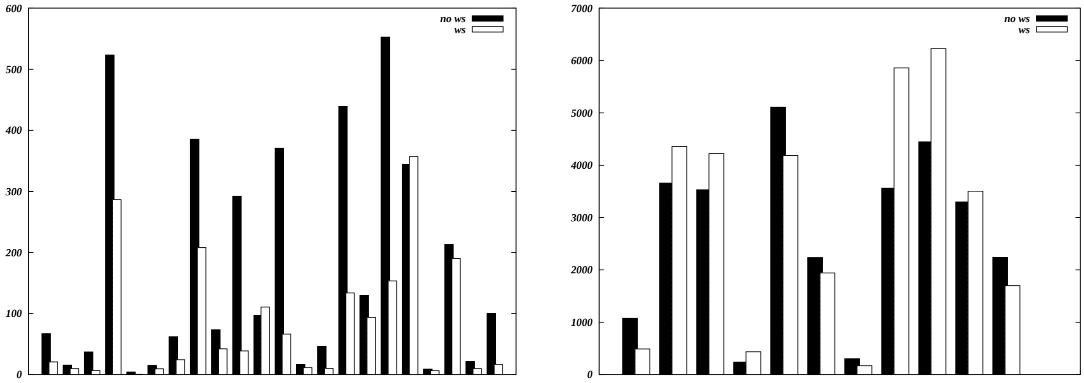


Figure 4.22: Warm-starting for VRP: $\alpha = 5\%, \beta = \pm 20\%$.

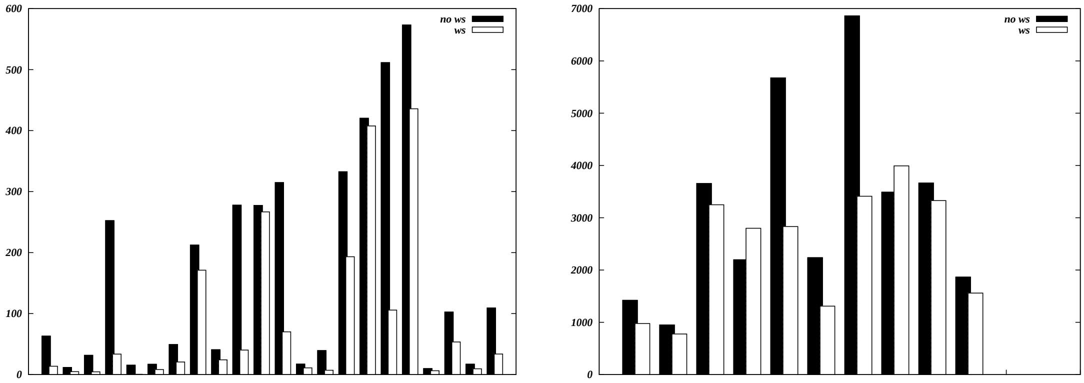


Figure 4.23: Warm-starting for VRP: $\alpha = 10\%, \beta = \pm 5\%$.

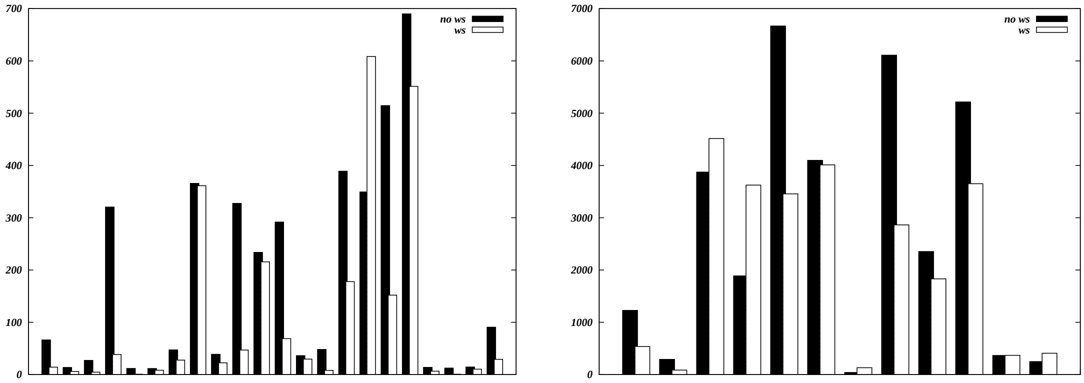


Figure 4.24: Warm-starting for VRP: $\alpha = 10\%, \beta = \pm 10\%$.

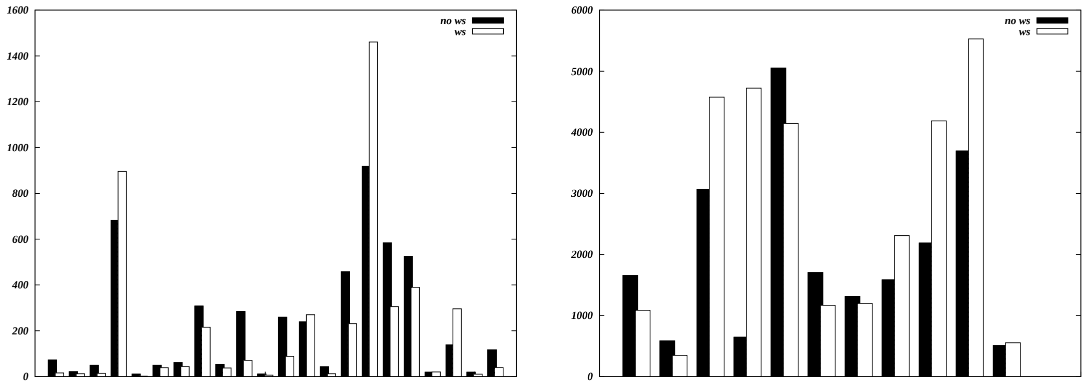


Figure 4.25: Warm-starting for VRP: $\alpha = 10\%, \beta = \pm 20\%$.

solution than x^*.

The motivation for warm starting in RINS is that if during the solution procedure RINS is being called repeatedly, then the sub-MILPs being solved will be similar. Our set-up for this experiment including only binary problems is as follows: we invoke RINS at the nodes for which the size of the search neighborhood is considered small enough to solve the restricted MILP with a branch-and-bound algorithm within a given time limit. Initially, we save the root node (including all the cuts) of the original problem as the base warm-start for the first RINS call; for the subsequent RINS calls, however, we use the complete tree or subtrees of the last RINS solve.

In order to get a valid partition from the branching tree of current RINS solve for the restricted MILP to be solved at next iteration, we apply the following simple rule on variable bounds. Let $(l_i)_j, (l_i)_{j+1}$ and $(u_i)_j, (u_i)_{j+1}$ be the lower and upper bounds of variable x_i for the RINS iterations j and $j+1$. Then, unless $(l_i)_j = (l_i)_{j+1}$ and $(u_i)_j = (u_i)_{j+1}$, we let $(l_i^t)_{j+1} = (l_i)_{j+1}$ and $(u_i^t)_{j+1} = (u_i)_{j+1}$ for each leaf node t of the warm-start tree.

We have tested our procedure for all mixed binary instances in MIPLIB3 (Bixby et al. [1998]), MIPLIB2003 (Achterberg et al. [2009]) and UNIBO (Mittelmann [2009b]). We compare 4 runs: default RINS and warm starting using nodes below the $r\%$ level of the tree, $r \in \{10, 50, 100\}$ with total time limit of 2 hours and time limit of 100 seconds for each RINS call.

Average behavior over all instances is

	Def	r=10	50	100
Avg. $\frac{\text{Time-per-call-Default}}{\text{Time-per-call}}$	1.0	2.53	2.46	2.59
Total # heuristic solutions	158	199	184	210

As can be seen in Figure 4.26, time spent in RINS is reduced dramatically, although only minor improvements are observed in total running time and optimality gap. Note that this behavior is not unexpected since only upper-bounds are affected by RINS. In Table 4.2, we consider improvements observed on particular instances mkc, sp98ar, aflow30a and swath for warm starting from different levels of the search tree. Since it is called more times within the allocated time, it helps to achieve either a better upper bound eventually or the same upper bound faster compared to solving the instances from scratch at each RINS call.

4.5.5 Stochastic Programming

Caroe and Schultz [1999] proposes a Lagrangian-based dual decomposition algorithm to solve two-stage stochastic pure integer programs. The version of SIP problem we consider in this section is with fixed, relatively complete, integer recourse with scenarios defined not only on the right hand side of second stage problem but also on its objective function and the technology matrix:

$$v = \min \ \{cx + Q(x) : Ax \geq b, \ x \in X\} \, , \tag{4.19}$$

where $Q(x) = E_\xi \phi(h(\xi) - T(\xi)x)$ and $\phi(s) = \min\{q(\xi)y : Wy \leq s, y \in Y\}$. Here, c, b, A, W are known vectors and matrices with appropriate dimensions. The vector ξ is a random variable defined on some probability space and for each ξ, the vectors $q(\xi), h(\xi)$ and the technology matrix $T(\xi)$ have appropriate dimensions. The sets X and Y denote the restrictions that all variables $x \in X$ and $y \in Y$ are nonnegative while some of them are integer or binary. For each ξ, a *scenario* is the realization of the random variable $(q(\xi), h(\xi), T(\xi))$.

We will assume that we have $k < \infty$ scenarios. With the notation $(q^j(\xi), h^j(\xi), T^j(\xi))$ for

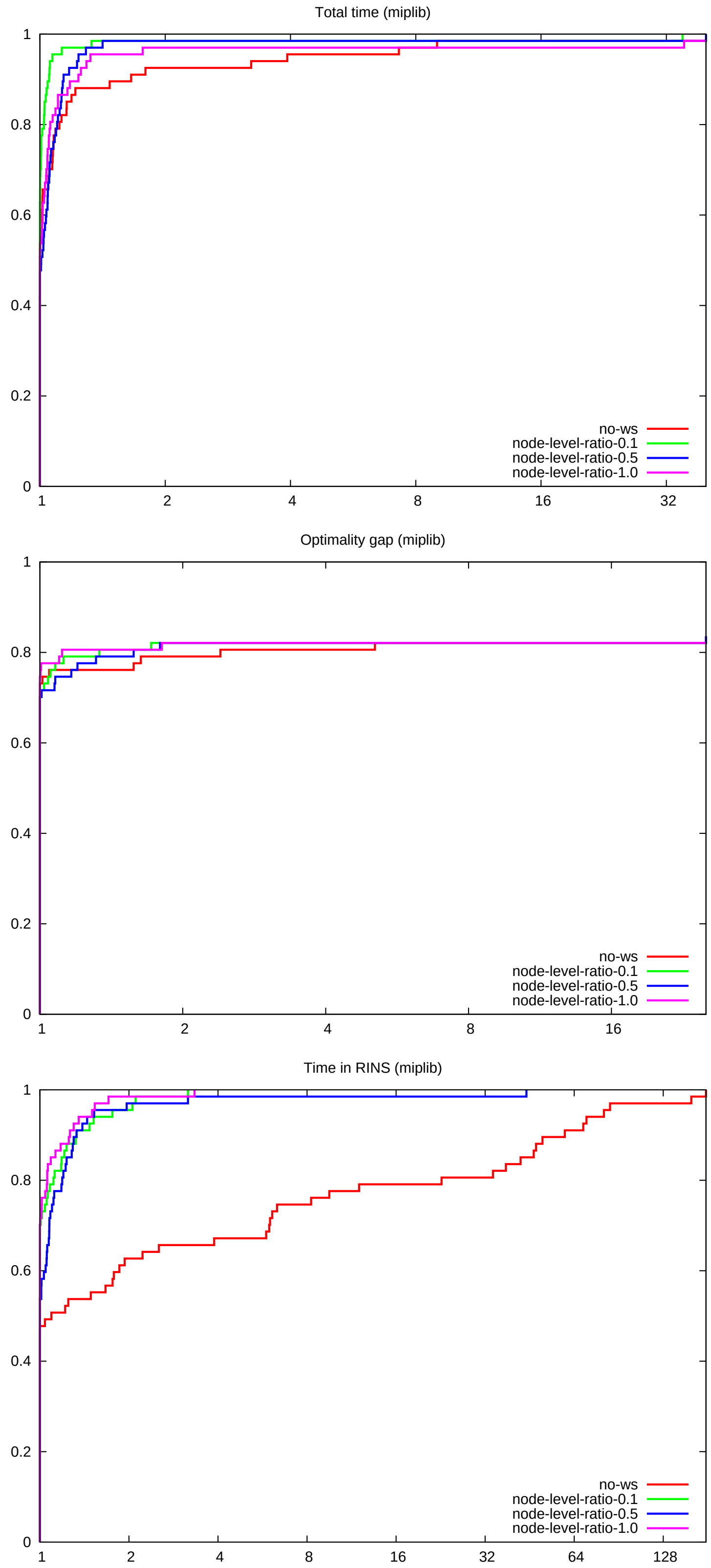


Figure 4.26: RINS - Warm-starting for MIPLIB Instances: Performance Profiles

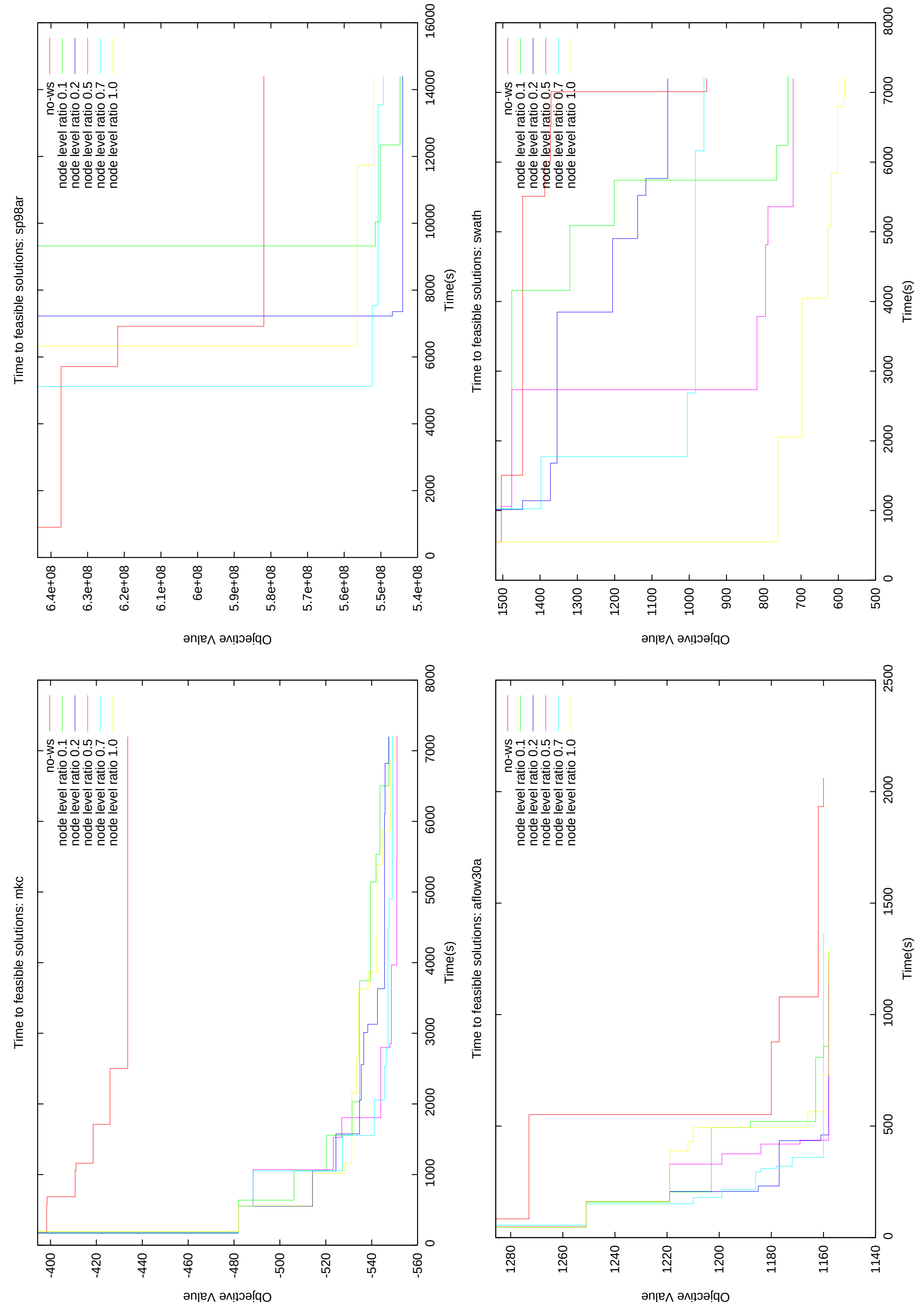


Table 4.2: RINS - Particular instances

each scenario $j = 1, ..., k$, if we further denote by p^j the probability of each scenario and define

$$S^j := \{(x, y^j) : Ax \geq b, x \in X, T^j x + W y^j \leq h^j, y^j \in Y\}, \tag{4.20}$$

then the deterministic equivalent of the problem can be written as

$$v = \min\{cx + \sum_j p^j q^j y^j : (x, y^j) \in S^j, \;\; j = 1, ..., k\} \tag{4.21}$$

The algorithm is based on *scenario decomposition*. That is, we can introduce the copies of first stage variables $x^1, ...x^r$ and rewrite the last equation as

$$v = \min \left\{ \sum_j p^j (cx^j + q^j y^j) : (x^j, y^j) \in S^j, j = 1, ..., k, x^1 = x^2 = ...x^k \right\} \tag{4.22}$$

The last set of equality conditions on these copies is called *non-anticipativity constraint* and states that the first stage decision should be independent of the second stage outcome. We The non-anticipativity constraint will be represented by

$$\sum_j H^j x^j = 0 \tag{4.23}$$

with appropriate dimensions.

Introducing the penalty vector u, the Lagrangian relaxation with respect to the non-anticipativity condition is the problem of finding $x^j, y^j, j = 1, ..., k$ such that:

$$D(u) = \min \;\; \{\sum_j L_j(x^j, y^j, u) : (x^j, y^j) \in S^j\} \tag{4.24}$$

where $L_j(x^j, y^j, u) = p^j(cx^j + q^j y^j) + u(H^j x^j), j = 1, ..., k$. Now, we have converted our initial problem to find

$$Z_{LD} = \max_u D(u) \tag{4.25}$$

Note that the main advantage of this formulation is that for any multiplier u, we can separate the problem into subproblems for each scenario

$$D(u) = \sum_{j}^{k} D_j(u) \, , \tag{4.26}$$

where for each $j = 1, ..., k$, $D_j(u) = \min \, \{L_j(x^j, y^j, u) : (x^j, y^j) \in S^j\}$ is an MILP subproblem. However, Caroe and Schultz [1999] showed that solving (4.26) does not guarantee to obtain the optimal solution to (4.21) due to existence of a duality gap. Therefore, they also presented a branch-and-bound procedure. We will not go into details but just note that, at each note t, we solve Lagrangian dual relaxation Z^t_{LD}, which is basically Z_{LD} added with branching decisions made on some components of x, to give a lower bound on Z_{LD} (see Section 2.2.3). The rest is the same with a regular branch-and-bound algorithm.

Table 4.3 shows the results of a set of two-stage stochastic programs from Ahmed [2004], Felt [2004], Holmes [2004]. We used a very straightforward implementation of subgradient algorithm to solve each Z^t_{LD}. Note that subgradient algorithm requires to solve a sequence of related MILPs. At root node for instance and at l^{th} iteration of subgradient optimization, the problems $D_j(u^l) = \min \, \{L_j(x^j, y^j, u^l) : (x^j, y^j) \in S^j\}, \; j = 1, ..., r$ are solved and at next iteration, $D_j(u^{l+1})$, for each j, will be differ from $D_j(u^l)$ only by some objective function coefficients. We used SYMPHONY to solve the subproblems with and without warm-starting from one iteration to the next. SUTIL (Linderoth [2006]) was used to read in the instances. The presence of a gap indicates that the problem was not solved to within the gap tolerance in the time limit. Although the running times are not competitive overall because of the slow convergence of our subgradient algorithm, one can clearly see the improvement arising from the use of warm-starting.

4.5.6 Bicriteria Problems

A *bicriterian MILP* can be generalized by

$$\begin{aligned} \text{vmin} \quad & [cx, qx] \\ \text{s.t.} \quad & x \in \mathcal{S}(b) \end{aligned} \tag{4.27}$$

Problem	Tree Size Without WS	Tree Size With WS	% Gap Without WS	% Gap With WS	CPU Without WS	CPU With WS
storm8	1	1	-	-	14.75	8.71
storm27	5	5	-	-	69.48	48.99
storm125	3	3	-	-	322.58	176.88
LandS27	71	69	-	-	6.50	4.99
LandS125	37	29	-	-	15.72	12.72
LandS216	39	35	-	-	30.59	24.80
dcap233_200	39	61	-	-	256.19	120.86
dcap233_300	111	89	0.387	-	1672.48	498.14
dcap233_500	21	36	24.701	14.831	1003	1004
dcap243_200	37	53	0.622	0.485	1244.17	1202.75
dcap243_300	64	220	0.0691	0.0461	1140.12	1150.35
dcap243_500	29	113	0.357	0.186	1219.17	1200.57
sizes3	225	165	-	-	789.71	219.92
sizes5	345	241	-	-	964.60	691.98
sizes10	241	429	0.104	0.0436	1671.25	1666.75

Table 4.3: Results of warm-starting to solve stochastic integer programs.

Solving (4.27) is the problem of generating the set of *efficient* solutions $\mathcal{E} = \{x_1, ..., x_k\}$ where for any $\bar{x} \in \mathcal{E}$, there does not exist a second distinct feasible solution $\hat{x}$ such that $c\hat{x} \leq c\bar{x}$ and $q\hat{x} \leq q\bar{x}$ and at least one inequality is strict. If both of the inequalities are strict, then $\bar{x}$ is called strongly efficient. The goal is to enumerate the set of all *Pareto outcomes* $\mathcal{K} = \{(cx, qx) \mid x \in \mathcal{E}\}$.

An extreme points on the lower side of the $conv(\mathcal{K})$ is called a *supported outcome* and can be obtained by solving the *weighted sum* problem

$$\min((1-\phi)c + \phi q)x \tag{4.28}$$

where $\phi \in [0, 1]$. Clearly, the set of all supported outcomes can be obtained by solving (4.28) for all $\phi \in [0, 1]$. Note that,

$$((1-\phi)c + \phi q)x = cx - \phi cx + \phi qx = cx + \phi(q - c)x \tag{4.29}$$

and therefore, enumerating all the supported solutions is the same as solving the parametric objective MILP $z_c(\phi)$ with $\tilde{c} = q - c$.

On the other hand, the complete set of Pareto outcomes can be obtained by considering the *weighted Chebyshev norms* (WCN). Assume that $\mathcal{K}$ is *uniformly dominant* (that is, if all efficient solutions are strong). Let x^c be a solution to a weighted sum problem with $\phi = 0$ and x^d be a solution with $\phi = 1$ (these are called the *utopia* solution values). Then, for a given $\phi \in [0, 1]$, the WCN problem is

$$\begin{aligned} \min_x \max \quad & \{((1-\phi)(cx - cx^c),\ \phi(qx - qx^q))\} \\ \text{s.t.} \quad & x \in S(b) \end{aligned} \tag{4.30}$$

and any solution to this problem corresponds to a Pareto outcome. Note that (4.30) can easily be linearized to an MILP by introducing an additional variable and $\mathcal{K}$ can be produced by solving this MILP for all $\phi \in [0, 1]$. Such an algorithm is recently presented by Ralphs et al. [2006]. We will not get into details but note that, as in the parametric analysis of MILP, they generate a set of candidate ϕ values and solve a corresponding sequence of MILP instances.

SYMPHONY has a function as of the implementation of this algorithm that can be invoked through **sym_mc_solve()** (or **multiCriteriaBranchAndBound()** in OSI case). We have added the warm-starting capability to the supported outcome generator since in this case only the objective varies and cuts remain valid for each instance. We implemented four options:

1. Use the branch-and-cut tree of the first utopia solution as an initial base and warm start a subsequent MILP instance using the branch-and-cut tree of the previous instance.

2. Keep the branch-and-cut trees of the utopia solutions as fixed bases and warm start a subsequent MILP instance with parameter ϕ using the first base if $\phi \leq 0.5$, and the second base otherwise.

3. Keep the branch-and-cut trees of the utopia solutions as bases and warm start a subsequent MILP instance with parameter ϕ using the first base if $\phi \leq 0.5$, and the second base otherwise. The bases are not fixed and are updated with the search tree of subsequent MILPs.

4. Keep the branch-and-cut tree of each MILP instance as potential bases. To warm start a subsequent instance, compare the ϕ parameters and choose the closest base.

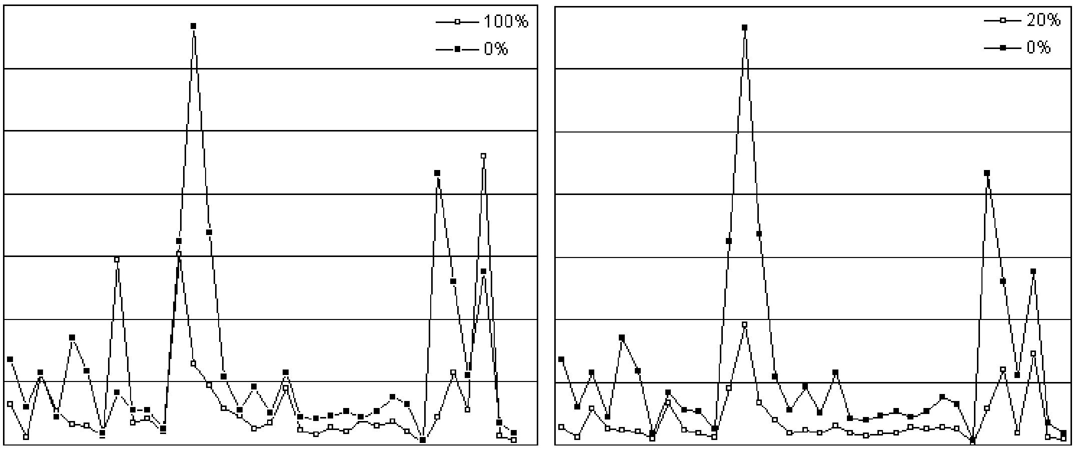


Figure 4.27: Results of warm-starting to solve bicriteria optimization problems.

We have used bicriteria solver for the instances (Capacitated Network Routing Problems) of size 15 reported in Ralphs et al. [2006]. For this experiment, we used option (4) and chose the warm-start tree to be the first $\upsilon\%$ of the nodes generated during the solution algorithm of chosen base. In Figure 4.27, the first chart compares υ values of 0 and 100, whereas the second chart compares υ values of 0 and 20. The results with $\upsilon = 20$ show a clear and marked improvement over those with $\upsilon = 0$. Results with $\upsilon = 100$ show a noticable effect in some cases, but an overall improvement.

4.5.7 Sensitivity Analysis

In Table 4.4, we report the results of testing the code of Figure 4.6 on the MIPLIB3 (Bixby et al. [1998]) file **flugpl**. Note that, for each pair of right-hand side values, the smaller of the two numbers (shown above) is the lower bound obtained by SYMPHONY, whereas the larger number is the optimal solution value for the problem with the given right-hand sides values.

On the other hand, bicriteria solver can also be used for complete parametric analysis of a single objective coefficient. Assume that for a given $k \in N$, we want to find the value of

$$\begin{aligned} Z_k(\varphi) = \quad & \textstyle\sum_{j \neq k} c_j x_j + \varphi x_k \\ \text{s.t.} \quad & x \in S(b), \end{aligned} \tag{4.31}$$

	6000	6500	7000	7500	8000	8500	9000	9500	10000
7000	1109789.89 1165500.00	1118201.35 1165500.00	1126612.81 1165500.00	1135024.27 1168500.00	1146369.28 1183500.00	1151847.18 1194000.00	1160258.64 1194000.00	1168670.10 1194000.00	1177081.56 1200000.00
7500	1115727.39 1165500.00	1124138.85 1165500.00	1132550.31 1165500.00	1140961.77 1168500.00	1154383.33 1183500.00	1157784.68 1197000.00	1166196.14 1197000.00	1174607.60 1197000.00	1183019.06 1203000.00
8000	1121664.89 1165500.00	1130076.35 1165500.00	1139383.33 1165500.00	1154383.33 1168500.00	1169383.33 1183500.00	1169450.00 1198500.00	1172133.64 1212000.00	1180545.10 1212000.00	1188956.56 1218000.00
8500	1127602.39 1168500.00	1136013.85 1168500.00	1150764.07 1168500.00	1165764.07 1171500.00	1184383.33 1186500.00	1184470.59 1201500.00	1184470.59 1227000.00	1186482.60 1227000.00	1194894.06 1233000.00
9000	1133539.89 1183500.00	1143960.00 1183500.00	1158960.00 1183500.00	1173960.00 1186500.00	1201500.00 1201500.00	1201500.00 1216500.00	1201500.00 1240500.00	1201500.00 1240500.00	1201500.00 1246500.00
9500	1139477.39 1198500.00	1147888.85 1198500.00	1162245.00 1198500.00	1177245.00 1201500.00	1201517.78 1216500.00	1201527.78 1231500.00	1201527.78 1240500.00	1201527.78 1240500.00	1206769.06 1246500.00
10000	1145414.89 1246500.00	1153826.35 1246500.00	1162690.00 1246500.00	1177690.00 1246500.00	1201517.78 1246500.00	1201527.78 1246500.00	1201527.78 1246500.00	1204295.10 1246500.00	1212706.56 1252500.00
10500	1151352.39 1261500.00	1159763.85 1261500.00	1168175.31 1261500.00	1182987.50 1261500.00	1201517.78 1261500.00	1201527.78 1261500.00	1201821.14 1261500.00	1210232.60 1261500.00	1210232.60 1267500.00
11000	1157289.89 1276500.00	1165701.35 1276500.00	1174112.81 1276500.00	1186517.78 1276500.00	1201517.78 1276500.00	1201527.78 1276500.00	1207758.64 1276500.00	1216170.10 1276500.00	1224581.56 1282500.00

Table 4.4: Sample results with SYMPHONY's sensitivity analysis function

for all $\varphi \in \mathbb{R}$. Consider the bicriteria problem

$$\begin{aligned} \text{vmin} \quad & [\textstyle\sum_{j \neq k} c_j x_j, x_k] \\ \text{s.t.} \quad & x \in \mathcal{S}(b). \end{aligned} \tag{4.32}$$

Note that solving (4.32) to enumerate the supported solutions is the same as solving

$$\begin{aligned} \vartheta(\phi) = \quad & \phi \textstyle\sum_{j \neq k} c_j x_j + (1 - \phi) x_k \\ \text{s.t.} \quad & x \in S(b) \end{aligned} \tag{4.33}$$

for all $\phi \in [0, 1]$. However, it is easy to see that by setting $\varphi = \frac{1-\phi}{\phi}$,

$$Z_k(\varphi) = (1 + \varphi)\vartheta(\frac{1}{1 + \varphi}) \tag{4.34}$$

As an illustration of this type of sensitivity analysis, we applied the code in Figure 4.28 to the following simple ILP and got the value function $Z_2(\varphi)$ as shown in Table 4.5.

$$\begin{aligned} \min \quad & -8x_1 + \varphi x_2 \\ \text{s.t.} \quad & 7x_1 + x_2 \leq 56 \\ & 28x_1 + 9x_2 \leq 252 \\ & 3x_1 + 7x_2 \leq 105 \\ & x_1, x_2 \geq 0, \text{integral} \end{aligned} \tag{4.35}$$

```
int main(int argc, char **argv)
{
   OsiSymSolverInterface si;
   si.parseCommandLine(argc, argv);
   si.loadProblem();
   si.setObj2Coeff(0, 1);
   si.multiCriteriaBranchAndBound();
}
```

Figure 4.28: Performing sensitivity analysis with SYMPHONY's bicriteria solver.

φ range	$Z_2(\varphi)$	x_1^*	x_2^*
$(-\infty, -16.000)$	15φ	0	15
$(-16.000, -8.000)$	$-32 + 13\varphi$	4	13
$(-8.000, -2.667)$	$-40 + 12\varphi$	5	12
$(-2.667, -1.333)$	$-56 + 6\varphi$	7	6
$(-1.333, \infty)$	-64	-8	0

Table 4.5: Price function for a simple ILP

Chapter 5

Conclusions and Future Research

The basic theory behind some of the ideas presented here was developed more than three decades ago and it would seem that little progress had been made towards a practical framework after the discovery of this initial theory. We believe, however, that the methods discussed here represent the clearest path to bringing a practical notion of duality for mixed integer linear programs to fruition, since these techniques work in tandem with algorithms that are already effective in solving the primal problem. To be more specific, we first summarize the results of this study and then discuss some of the directions one might explore in the future to continue this work.

5.1 Conclusions

In this thesis, we first presented a survey of existing theory and methodology for constructing dual functions. From the standpoint of computational practice, the importance of these methodologies is that they allow us to extend to the realm of MILP some of the useful techniques already well-developed for linear programming, such as approximating the value function in applications, the ability to perform post-solution sensitivity analyses, and the ability to warm start solution processes of a modified problem. In Section 2.2.5, we showed that it is possible to obtain dual functions from the branch-and-cut algorithm. The importance of obtaining such dual functions is clear from the fact that the branch-and-cut method is the most common method in practice for solving MILPs.

In Chapter 3, we presented our theoretical results related to the structure of the value function of an MILP with a single constraint and outlined a complete procedure to evaluate the value function itself. It does seem that practical methods may be possible, since the enumeration required to evaluate the value function for a new right-hand side is reasonable in cases where the coefficients and right-hand side are small in absolute value. It is clear that a number of these results can be extended to more general cases, but such extension seems unlikely to yield any practical methods. Nevertheless, we believe that the results for the single-constraint case presented herein have added value to our knowledge of understanding the general case and have more potential of being used for analyzing the general value function.

On the other hand, we described possible ways of constructing approximations of the value function of an MILP that can be substituted for the value function in applications, just as in the LP case. We presented an alternate way of generating dual functions for general MILPs using its single row approximations without the need of a primal solution algorithm. It is apparent that these ideas enable us to derive, from the computational view, more efficient algorithms than those evaluating the value function explicitly. We illustrated in Section 3.4 such a usage of these approximations for large-scale/multi-stage problems. It is our hope that this work will provide the foundation for developing practical solution methods to solve large instances that depend on the value function of a general MILP.

Finally in Chapter 4, we developed procedures for both sensitivity analysis and warm starting using the dual information obtained through the primal solution algorithms. The computational experiments reveal that retaining the wasted information produced in solution process of an MILP and using it effectively might speed up the solution time of a related MILP such that the gain in some cases would be more than a factor of five. As we discussed, these methodologies can further be integrated efficiently for the special applications where the underlying structure reveal more information. We believe that our work in this area has a great potential to be beneficial considering the fact that many optimization algorithms in real-time environments require the solution of a sequence of related MILPs.

5.2 Future Research

We are cognizant of the fact that many more open questions remain. One of those is to delve more into the structure of the value function of an MILP and to determine whether the procedures suggested here to obtain the value function of an MILP with a single constraint might be extended to the general case in a practical way. A related, and perhaps more important, future path, is to seek more efficient algorithms for the subproblems to be solved at each iteration of the dual methods proposed in Section 3.4 for large-scale problems. Parallel to that, it is also not known how to obtain better-structured approximation of the value function that can be easily integrated with current optimization techniques. Finally, even though we have described possible ways of selecting a good warm-start tree to initiate the solution procedure of a modified instance, the quest of determining the best one remains unchallenged. In this sense, it is clear that there are many opportunities to be explored to improve the warm-starting techniques presented in this study.

We are aware that some of the results presented here fall short of being practical for obvious reasons when compared to the LP case. We do, however, believe that this is a fruitful direction for future research and there are some obvious advances that seem close at hand.

Bibliography

T. Achterberg, T. KCOH, and A. MARTIN. MIPLIB2003, Available from http://miplib.zib.de/miplib2003.php, 2009.

S. Ahmed, 2004. SIPLIB, Available from http://www.isye.gatech.edu/ sahmed/siplib.

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

M. Barton and R. Laatsch. Maximal and minimal subadditive extensions. *Amer. Math. Monthly*, 73(4, part II):141–143, 1966.

A.I. Barvinok. Computing the volume, counting integral points, and exponential sums. *Discrete Comput. Geom.*, 10:1–13, 1993.

A.I. Barvinok. Polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math of Operations Research*, 19:769–779, 1994.

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

R. E. Bixby, S. Ceria, C. M. McZeal, and M. W.P. Savelsbergh. An updated mixed integer programming library: Miplib 3. *Optima*, 58:12–15, 1998.

R.E. Bixby, E.A. Boyd, and R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.

C.E. Blair. Minimal inequalities for mixed integer programs. *Discrete Mathematics*, 24:147–151, 1978.

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

C.E. Blair. Sensitivity analysis for knapsack problems: A negative result. *Discrete Applied Mathematics*, 81(1–3):133–139, 1998.

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

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

C.E. Blair and R.G. Jeroslow. The value function of an integer program. *Mathematical Programming*, 23:237–273, 1982.

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

M. Brion and M. Vergne. Residue formulae, vector partition functions and lattice points in rational polytopes. *Journal of the American Mathematical Society*, 10:797–833, 1997.

K. Leyton Brown, M. Pearson, and Y. Shoham. Cats - combinatorial auction test suite, version 2.0. 2007. Available from http://www.cs.ubc.ca/ kevinlb/CATS.

A. Bruckner. Minimal superadditive extensions of superadditive functions. *Pacific J. Math.*, 10: 1155–1162, 1960.

C. Burdet and E.L. Johnson. A subadditive approach to solve integer programs. *Annals of Discreet Mathematics*, 1:117–144, 1977.

C.C. Caroe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters*, (24):37–54, 1999.

COIN-OR. Computational infrastructure for operations research. 2009. Available from http://www.coin-or.org.

E. Danna, E. Rothberg, and C. Le Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102:71–90, 2005.

M. Eso. Parallel branch and cut for set partitioning. *Ph.D. Dissertation, Cornell University*, 1999.

P. Feautrier. Parametric integer programming. *RAIRO Recherche Operationnelle*, 22(3):243–268, 1988.

A. Felt. Stochastic linear programming data sets, 2004. Available from http://www.uwsp.edu/math/afelt/slptestset.html.

M. L. Fisher. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27:1–17, 1981.

A.M. Geoffrion and R.M. Nauss. Parametric and postoptimality analysis in integer.linear programming. *Management Science*, 23(5):453–466, 1977.

R.E. Gomory. Some polyhedra related to combinatorial problems. *Linear Algebra and Its Applications*, 2:451–558, 1969.

R.E. Gomory and E.L. Johnson. Some continuous functions related to corner polyhedra, II. *Mathematical Programming*, 3:359–389, 1972.

D. Holmes. Stochastic linear programming data sets, 2004. Available from http://users.iems.nwu.edu/ jrbirge/html/dholmes/post.html.

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 Mathematical Programming, T.C. Hu and S.M. Robinson (eds.) Academic Press, New York, NY*, pages 185–211, 1973.

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

E.L. Johnson. On the group problem and a subadditive approach to integer programming. *Annals of Discreet Mathematics*, 5:97–112, 1979.

D. Klabjan. A new subadditive approach to integer programming: Theory and algorithms. In *Proceedings of the Ninth Conference on Integer Programming and Combinatorial Optimization*, pages 384–400, Cambridge,MA, 2002.

D. Klein and S. Holm. Integer programming post-optimal analysis with cutting planes. *Management Science*, 25(1):64–72, 1979.

N. Kong, A.J. Schaefer, and B. Hunsaker. Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Math. Programming, Ser. B*, pages 1–22, 2006.

R. G. Laatsch. Extensions of subadditive functions. *Pacific J. Math.*, 14:209–215, 1964.

J.B. Lasserre. Duality and farkas lemma for integer programs. In *Optimization : Structure and Applications, E. Hunt and C.E.M. Pearce, eds.,*, Applied Optimization Series, Kluwer Academic Publishers, 2009.

J.B. Lasserre. Generating functions and duality for integer programs. *Discrete Optimization*, 2(1): 167–187, 2005a.

J.B. Lasserre. Integer programming, duality and superadditive functions. *Contemporary Mathematics*, 374:139–150, 2005b.

Jean B. Lasserre. The integer hull of a convex rational polytope. *Discrete & Computational Geometry*, 32(1):129–139, 2004.

M. Libura. Sensitivity analysis for minimum Hamiltonian path and traveling salesman problems. *Discrete Applied Mathematics*, 30(2):197–211, 1991.

M. Libura. Optimality conditions and sensitivity analysis for combinatorial optimization problems. *Control and Cybernetics*, 25(6):1165–1180, 1996.

J.T. Linderoth. SUTIL, Available from http://coral.ie.lehigh.edu/sutil/index.html, 2006.

D. C. Llewellyn and J. Ryan. A primal-dual integer programming algorithm. *Discrete Applied Mathematics*, 45:261–276, 1993.

J. De Loera, D. Haws, R. Hemmecke, P. Huggins, B. Sturmfels, and R. Yoshida. Short rational functions for toric algebra and applications. *J. Symbolic Comp.*, 38:959–973, 2004a.

J. De Loera, D. Haws, R. Hemmecke, P. Huggins, and R. Yoshida. Three kinds of integer programming algorithms based on barvinok's rational functions. *10th International IPCO Conference. Lecture Notes in Computer Science*, 3064:244–255, 2004b.

E. Loukakis and A.P. Muhlemann. Parameterisation algorithms for the integer linear programs in binary variables. *European Journal of Operational Research*, 17:104–115, 1984.

A. Makhorin. Introduction to glpk. 2009. Available from http://www.gnu.org/software/glpk/glpk.html.

R.E. Marsten and T.L. Morin. Parametric integer programming: the right-hand side case. *Annals of. Discr. Math*, 1:375–390, 1977.

R.R. Meyer. Integer and mixed-integer programming models: General properties. *Journal of Optimization Theory and Applications*, 16(3/4):191–206, 1975.

H. Mittelmann. Mixed Integer Linear Programming Benchmark, http://plato.asu.edu/ftp/milpf.html, 2009a.

H. Mittelmann. MILP Test Set, Available from http://plato.asu.edu/ftp/unibo/ , 2009b.

R.M. Nauss. *Parametric Integer Programming*. 1975. Ph.D. Dissertation, available as a Working Paper No. 226, Western Management Science Institute, UCLA, January.

R.M. Nauss. *Parametric Integer Programming*. University of Missouri Press, Columbia, Missouri, 1979.

G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1988.

Y. Ohtake and N. Nishida. A branch-and-bound algorithm for 0-1 parametric mixed integer programming. *Operations Research Letters*, 4(1):41–45, 1985.

C.J. Piper and A.A. Zoltners. Some easy postoptimality analysis for zero-one programming. *Management Science*, 22(759–765), 1976.

T.K. Ralphs. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing*, (29): 607–629, 2003.

T.K. Ralphs. Symphony version 5.0 user's manual. *Lehigh University Industrial and Systems Engineering Technical Report 04T-020*, 2004.

T.K. Ralphs. Parallel branch and cut. In *Parallel Combinatorial Optimization, E.Talbi, ed.*, pages 53–101, 2006.

T.K. Ralphs and M. Güzelsoy. Duality and warm starting in integer programming. In *Proceedings of the 2006 NSF Design, Service, and Manufacturing Grantees and Research Conference*, 2006.

T.K. Ralphs and M. Güzelsoy. The symphony callable library for mixed integer programming. *Proceedings of the Ninth INFORMS Computing Society Conference*, pages 61–76, 2005.

T.K. Ralphs, L. Kopman, W.R. Pulleyblank, and L.E. Trotter. On the capacitated vehicle routing problem. *Mathematical Programming*, (94):343–359, 2003a.

T.K. Ralphs, L. Ladányi, and M.J. Saltzman. Parallel branch, cut, and price for large-scale discrete optimization. *Mathematical Programming*, (98):253–280, 2003b.

T.K. Ralphs, M.J. Saltzman, and M.M. Wiecek. An improved algorithm for biobjective integer programming. 2006.

G.M. Roodman. Postoptimality analysis in zero-one programming by implicit enumeration. *Naval Research Logistics Quarterly*, 19:435–447, 1972.

G.M. Roodman. Postoptimality analysis in zero-one programming by implicit enumeration: The mixed integer case. *The Amos Tuck School of Business Administration, Darthmouth College*, October 1973.

S. Rountree and B. Gillett. Parametric integer linear programming: A synthesis of branch and bound with cutting planes. *European Journal of Operational Research*, 10:183–189, 1982.

L. Schrage and L.A. Wolsey. Sensitivity analysis for branch and bound integer programming. *Operations Research*, 33(5):1008–1023, 1985.

A. Schrijver. On cutting planes. *Annals of Discrete Mathematics*, 9:291–296, 1980.

A.S. Schulz. On the relative complexity of 15 problems related to 0/1-integer programming. In *Research Trends in Combinatorial Optimization, W.J. Cook, L. Lovsz, J. Vygen, eds.*, pages 399–428, Springer, Berlin, 2009.

J.F. Shapiro. Sensitivity analysis in integer programming. *Annals of Discrete Mathematics*, 1: 467–477, 1977.

Y.N. Sotskov, V.K. Leontev, and E.N. Gordeev. Some concepts of stability analysis in combinatorial optimization. *Discrete Applied Mathematics*, 58(2):169–190, 1995.

VRPLIB. Vehicle Routing Problems Test Set, Available from http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data, 2006.

L.A. Wolsey. The b-hull of an integer program. *London School of Economics and CORE*, 1979.

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

Biography

Menal Guzelsoy is a doctoral candidate in the Industrial and Systems Engineering Department at Lehigh University. He received his B.S in Industrial Engineering from Bilkent University, Turkey. His research interests include both theoretical and computational aspects of duality, post-optimal analysis and warm-starting for mixed integer linear programs. He is the current developer of SYMPHONY optimization solver and an active member of COIN-OR Foundation and Computational Optimization Research at Lehigh (COR@L) Lab.