Representing Integer Program Value Functions with Neural Networks

Tu Anh-Nguyen[*], Joey Huchette[†], Andrew J. Schaefer[‡]

October 22, 2024

Abstract

We study the value function of an integer program (IP), which characterizes how the optimal objective value changes as the right-hand sides vary. We show that the IP value function can be approximated to any desired degree of accuracy using machine learning (ML) techniques. First, drawing on the well-known equivalency between IP value functions and Chvátal-Gomory (CG) functions, we show that there exists a neural network (NN) that approximates the IP value function to an arbitrary degree of accuracy and whose size scales exponentially in the CG rank of the IP value function and logarithmically in the magnitude of the data. This approach requires an explicit representation of the IP value function as a CG function, which motivates our second NN architecture designed to emulate the fundamental operations comprising CG functions. We show that 1) the function represented by the second architecture is monotonically non-decreasing and superadditive, 2) approximations using the second architecture implicitly yield CG inequalities of the IP, and 3) we can produce an exact representation of the IP value function by solving a mixed-integer programming problem.

1 Introduction

Duality is a fundamental concept in optimization, and while linear and convex programming duality are widely understood and critical for modern algorithms, less is known about integer programming (IP) duality. It has been shown that the optimal solution of *superadditive* IP duality - a strong dual for IP - is the IP value function [44]. The seminal work of Blair and Jeroslow characterized the value function of a maximization IP, parameterized by its right-hand side, as a Chvátal-Gomory (CG) function, meaning that it can be constructed recursively from the set of linear functions through the composition of addition, nonnegative multiplication, minimization, and rounding down operations [9, 10, 12]. Despite its elegance, this result has a surprisingly limited impact on IP computation.

The value function is crucial in solving large-scale linear or integer programs as it is a fundamental tool for decomposition methods such as Benders' decomposition [5, 35] or the Integer L-shaped Algorithm [14]. However, an effective method for finding the IP value function remains elusive. Recently, researchers have computed it exactly over a bounded domain. Kong et al. [26] and Trapp et al. [40] computed the IP value function on a bounded set of integral points and took advantage of IP duality to reduce computation at some lattice points. In a similar line of work, Tavaslıoğlu et al. [39] extended the approach to the mixed-integer program (MIP) value function. These approaches stored the value functions directly or indirectly at every integral point in the bounded domain, thus limiting their applicability.

[*]Rice University. Email: `tu.na@rice.edu`
[†]Google. Email: `joehuchette@google.com`
[‡]Rice University. Email: `@rice.edu`

Recently, machine learning (ML) has received interest in the area of mathematical optimization [7]. Current research efforts in applying ML to discrete optimization often concentrate on enabling ML to make algorithmic decisions, such as node and branching selection in branch-and-bound algorithms [2, 21], or cut identification and classification [25, 37]. Alternatively, some research uses ML to directly estimate the solution of an IP from its input parameters [13, 27, 28, 43]. Furthermore, deep learning has frequently been utilized to approximate value functions for dynamic programming problems in reinforcement learning, neuro-dynamic programming, or Markov decision processes [8, 40]. Neural Networks (NNs) have been recently shown to be useful in approximating the mixed-integer programming (MIP) value function and heuristically solving stochastic programs [17, 29, 34].

NNs are typically constructed through the recursive application of basic operations, such as affine transformations and piecewise linear activation functions (e.g., ReLU or max pooling). With sufficient layers and neurons in each layer, NN are universal approximators, in the sense that they can closely approximate any continuous function [4, 20, 22, 30]. Despite this, it is often of practical utility to introduce bias or special architectures to a NN in certain applications, e.g., convolutional layers are used in image processing tasks [31], a recurrent NN or attention layer is one of the main components in natural language processing [18, 38].

IP value functions represent a well-defined class of functions with distinct properties. These functions are characterized by *discontinuity*, where small changes in input can lead to large changes in output; *monotonicity*, and *superadditivity*, where the function's value for the summation of vectors is at least the sum of its values for each vector. These special features present a distinct modeling challenge. Hence, this raises an intriguing question: Can NNs effectively model IP value functions? This paper answers this question affirmatively, demonstrating that NNs can indeed capture the complex and nuanced characteristics of IP value functions.

1.1 Summary of Contributions

Blair and Jeroslow's characterization of the IP value function exposes fundamental similarities between NNs and CG operations, which we summarized in Table 1.

Operations	**CG Function**	**Neural Network**
Summation	Linear Combination	Affine Combination
Multivariate Nonlinearity	Minimum over multiple inputs	Max-Pooling
Univariate Nonlinearity	Round-down	ReLU

Table 1: Analogies between CG Functions and NN Operations.

In this work, inspired by the similarity between CG and NN operations, we introduce two representation theorems that characterize how, and how well, NNs can approximate IP value functions.

Tree Representation Theorem. Consider an IP value function constructed using at most r CG operations (see Table 1), defined on a bounded domain $\mathcal{D}$. There exists a feed-forward ReLU NN whose number of neurons grows exponentially with r and logarithmically with the input domain's size and the coefficients of the IP value function, that approximates the IP value function within given $\epsilon > 0$ in L_1-norm over $\mathcal{D}$.

The Tree Representation Theorem offers valuable insights, primarily demonstrating the efficacy of NNs in approximating IP value functions. The theorem links the complexity of these functions with the requisite size of the NN needed for their approximation. Specially, *CG rank* gives a fundamentally different way to characterize complexity than generic universal approximation results, and so might offer tighter bounds. In contrast with modern ML models, which have proven to be universal approximators [4, 20, 22, 30], the architecture in the Tree Representation Theorem only searches in the space of CG functions and can potentially require less data for training [41]. However, a limitation of the Tree Representation Theorem is that it depends on the number and order of the CG operations in an IP value function, in addition to bounded but unknown coefficients. The forthcoming second representation theorem aims to address this gap.

Block Representation Theorem. Any IP value function of rank r (see Definition 5) on a domain $\mathcal{D}$ (not necessarily bounded) can be represented by a NN architecture with $O(r)$ layers, each with a bounded number of neurons (independent of r) with round-down activation functions.

The remainder of the paper is organized as follows. Section 2 provides structural results, which are the keys to proving the two representation theorems. Section 3 proves the existence of a NN that can approximate any IP value function within a given error tolerance via the Tree Representation Theorem. Section 4 gives the formal statement and proof for the Block Representation Theorem, as well as demonstrates how the IP value function and CG inequalities can be computed via a MIP problem.

2 Structural Results

The NN constructions are based on the three CG operations that can recursively construct the IP value function [11]. The motivations for the models in later sections on computing IP value functions are inspired by the analogy between a CG function and a NN as summarized in Table 1. Before stating the main theorems, we spend this section describing results that support the NN Representation Theorems. One of our most important contributions in this section is the construction of the IP value function based on CG inequalities.

2.1 Preliminaries

In this work, we demonstrate the ability of a NN to represent the IP value function of the form:

$$
\begin{aligned}
z(b) \coloneqq \max\ & c^T x && \text{(1a)}\\
& Ax \le \beta && \text{(1b)}\\
& x \in \mathbb{Z}^n_+, && \text{(1c)}
\end{aligned}
$$

where $c \in \mathbb{R}^n$ is a fixed objective, and $A \in \mathbb{Z}^{m\times n}$ is a fixed integral constraint matrix with $(a^1, a^2, \ldots, a^n)$ denoting its columns. Denote $X(\beta) \coloneqq \{x \in \mathbb{Z}^n_+ | Ax \le \beta\}$ as the set of feasible points for a given β, and $\mathcal{D} = \{\beta \in \mathbb{Z}^m | X(\beta) \neq \emptyset\}$ as the set of right-hand side vectors that make a feasible IP. If for some right-hand side vector β, the problem is unbounded from above, we let $z(\beta) = +\infty$, while if the problem is infeasible, we let $z(\beta) = -\infty$. Throughout this paper, we use $IP(\beta)$ to denote the IP with the right-hand side vector β, and $LP(\beta)$ to denote its LP relaxation, i.e., the LP obtained by relaxing the integral constraint (1c) into $x \ge 0$. We further assume that for every $\beta \in \mathcal{D}$, $IP(\beta)$ has a finite optimal solution. This is not a strong assumption because if $z(\beta) = +\infty$ for some right-hand side β, then there exists $x^* \in \mathbb{Z}^n_+$ such that $Ax^* \le 0$ and $c^T x^* > 0$, which means $z(\beta) = +\infty$ for every $\beta \in \mathcal{D}$ [32].

Definition 1. *[44] **(CG inequality)**. For a fixed right-hand side vector $b \in \mathcal{D}$. A CG inequality with respect to the feasible region of $IP(b)$ is an inequality generated by the following two CG steps, which are defined as follows:*

1. *Select a non-negative vector $u \in \mathbb{R}^m_+$, known as the CG multiplier.*
2. *Construct a CG inequality $\sum_{j\in[\![n]\!]} \lfloor ua^j \rfloor x_j \le \lfloor ub \rfloor$.*

Denote the convex hull of $X(b)$ as $S(b)$. When all the integer variables of $IP(b)$ can be bounded, $S(b)$ can be described by a finite number of CG inequalities [36]. Furthermore, for every CG multiplier u, the inequality $\sum_{j\in[\![n]\!]} \lfloor ua^j \rfloor x_j \le \lfloor ub \rfloor$ is satisfied by each $x \in X(b)$. We also say that such an inequality is a *valid inequality*. By adding CG inequalities once at a time, we can recursively apply the CG steps.

Definition 2. *[44] **(CG inequality rank)**. For a fixed right-hand side vector $b \in \mathcal{D}$. The rank 0 CG inequalities are the valid inequalities of the feasible region of $LP(b)$, i.e., $\{x \in \mathbb{R}^n_+ | Ax \le b\}$. Let $\pi^b x \le \pi^b_0$ be* *a CG inequality of $S(b)$. We say that $\pi^b x \leq \pi^b_0$ is of rank r if $\pi^b x \leq \pi^b_0$ is not equivalent to or dominated by a non-negative linear combination of CG inequalities with rank smaller than $r-1$, but is equivalent to or dominated by a non-negative linear combination of CG inequalities obtained through applying r CG steps.*

Characterizing solutions and complexity of IPs relies on CG inequalities and their rank. Another crucial concept for understanding IPs is the CG function, which plays a key role in describing the superadditive dual of an IP.

Definition 3. *[9] **(CG function)**. The class $\mathcal{G}$ of m-dimensional CG function is the smallest class of functions that satisfies:*

(i) If $f(v) = \lambda\beta$ where $\lambda \in \mathbb{Q}^m$, then $f \in \mathcal{G}$,

(ii) If $f_1, f_2 \in \mathcal{G}$ and $\mu_1, \mu_2 \in \mathbb{Q}_+$, then $\mu_1 f_1 + \mu_2 f_2 \in \mathcal{G}$,

(iii) If $f \in \mathcal{G}$, then $\lfloor f \rfloor \in \mathcal{G}$,

(iv) If $f_1, f_2 \in \mathcal{G}$, then $\min\{f_1, f_2\} \in \mathcal{G}$.

Definition 4. *[9] **Carriers of CG functions**. For each function $f \in \mathcal{G}$, we define $\mathbb{C}(f)$, named the carriers set of f, inductively as follows:*

(i) If f is linear, then $f \in \mathbb{C}(f)$.

(ii) If f can be written as $\mu_1 f_1 + \mu_2 f_2$ where $\mu_1, \mu_2 \in \mathbb{Q}_+$, $f_1, f_2 \in \mathcal{G}$ and $f'_1 \in \mathbb{C}(f_1), f'_2 \in \mathbb{C}(f_2)$, then $\mu_1 f'_1 + \mu_2 f'_2 \in \mathbb{C}(f)$.

(iii) If $f' \in \mathbb{C}(f)$ then $f' \in \mathbb{C}(\lfloor f \rfloor)$.

(iv) If f can be written as $\min\{f_1, f_2\}$ where $f_1, f_2 \in \mathcal{G}$ and $f'_1 \in \mathbb{C}(f_1)$, $f'_2 \in \mathbb{C}(f_2)$, then $\min\{f'_1, f'_2\} \in \mathbb{C}(f)$.

By Blair and Jeroslow [9, Corollary 2.11], the carriers of CG functions contain exactly one element. Intuitively, the carrier of a CG function is a piecewise linear function obtained by removing all round-downs. Without loss of generality, we refer to this unique function as the carrier of a CG function. Since a carrier can be written as a minimum of a finite number of linear functions, we use *coefficients* of a CG function to refer to the coefficients of the linear function constructing its carrier. Another important aspect of a CG function, similar to CG inequalities, is the rank of a CG function.

Definition 5. *[9] **(Pre-rank & Rank)**. A CG function f has pre-rank zero if it is a linear function. It has pre-rank $(r+1)$ if there are functions f_1, f_2 of pre-rank r which satisfy at least one of these conditions:*

(i) $f = \mu_1 f_1 + \mu_2 f_2$ for some $\mu_1, \mu_2 \in \mathbb{Q}_+$,

(ii) $f = \min\{f_1, f_2\}$,

(iii) $f = \lfloor f_1 \rfloor$.

In general, a CG function can have several pre-ranks. Its rank is defined as its smallest pre-rank.

In the following subsection, we describe the mechanics of approximating the round-down function via a piecewise linear continuous function. This result plays a vital role in approximating the IP value function using NNs with continuous activation functions like ReLU. Next, we derive the connection between the CG multipliers and the IP value function, which is the key result for our block NN architecture in Section 4.

2.2 Approximation of the Floor Function

As we can see that the discontinuity of a CG function only comes from the round-down function, a natural question is how we can effectively approximate the round-down function in the context of traditional NNs, i.e., only using affine transformation and ReLU activation. Thus, we describe the mechanics we use to approximate the round-down function in this section. For a real number $\epsilon \in (0,1)$, we define a continuous function h_ϵ which approximates the round-down operator $\lfloor \cdot \rfloor$.

$$h_\epsilon(x) \coloneqq \begin{cases} \lfloor x \rfloor & \text{if } \lfloor x \rfloor + 1 - \epsilon \leq x \leq \lfloor x \rfloor + 1, \\ \frac{1}{\epsilon}x + (1 - \frac{1}{\epsilon})(\lfloor x \rfloor + 1) & \text{otherwise.} \end{cases} \tag{2}$$

Lemma 1. *For every $\epsilon \in (0,1)$, h_ϵ is a continuous function. Furthermore, for every $x \in \mathbb{R}$, we have*

$$\lim_{\epsilon \to 0} h_\epsilon(x) - \lfloor x \rfloor = 0.$$

Lemma 2. *Let $l < u$ be two non-negative integers and $0 < \epsilon < 1$, then*

$$\int_l^u \|h_\epsilon(x) - \lfloor x \rfloor\| dx = \epsilon(u-l)/2.$$

Lemma 3 gives a natural extension of Lemma 2 for the approximation of a composition of round-down and a piecewise linear function. For a piecewise linear function $g : D \subseteq \mathbb{R}^n_+ \to \mathbb{R}_+$ defined over a box domain $D \coloneqq [l,u]$, we partition D into $D = \cup_{i \in [\![t]\!]} D_i$, such that g is an affine function on each D_i. For a box domain D, and a function f defined on D, we define

$$\|f\|_1 = \int_D |f(x)| dx.$$

Lemma 3. *Let $g : D \subset \mathbb{R}^n \to \mathbb{R}$ be an affine piecewise function defined on a bounded box domain D, we have*

$$\lim_{\epsilon \to 0^+} \|h_\epsilon(g) - \lfloor g \rfloor\|_1 = 0.$$

Proof. We denote $\alpha^i x + \gamma_i$ as the affine function of $g(x)$ on domain D_i, where $\cup_{i \in [\![t]\!]} D_i = D$ is a partition of D. Assuming that, for some $i \in [\![t]\!]$, $\alpha^i_j \neq 0$ for every $j \in [\![n]\!]$, we have

$$\begin{aligned}
\int_{D_i} |h_\epsilon(g(x)) - \lfloor g(x) \rfloor| dx &= \int_{l_1}^{u_1} \cdots \int_{l_n}^{u_n} |h_\epsilon(\alpha^i \cdot x + \gamma_i) - \lfloor \alpha^i \cdot x + \gamma_i \rfloor| dx \\
&= \int_{\alpha^i_1 l_1}^{\alpha^i_1 u_1} \cdots \int_{\alpha^i_n l_n}^{\alpha^i_n u_n} |h_\epsilon(\mathbb{1} \cdot x + \gamma_i) - \lfloor \mathbb{1} \cdot x + \gamma_i \rfloor| dx \\
&\leq \prod_j |\alpha^i_j (u_j - l_j)| \int_{\alpha^i \cdot l}^{\alpha^i \cdot u} |h_\epsilon(t + \gamma_i) - \lfloor t + \gamma_i \rfloor| dt \\
&= \frac{1}{2} \prod_j |\alpha^i_j (u_j - l_j)| \epsilon(\alpha^i \cdot u - \alpha^i \cdot l).
\end{aligned}$$

Thus, by applying this inequality for every piece of D, we derive

$$\begin{aligned}
\int_D |h_\epsilon(g(x)) - \lfloor g(x) \rfloor| dx &= \sum_{t \in [\![t]\!]} \int_{D_i} |h_\epsilon(g(x)) - \lfloor g(x) \rfloor| dx \\
&\leq \frac{1}{2} \sum_{t \in [\![t]\!]} \prod_j |\alpha^i_j (u_j - l_j)| \epsilon(\alpha^i \cdot u - \alpha^i \cdot l) \\
&\leq \frac{1}{2} \epsilon \sum_{t \in [\![t]\!]} \prod_j |\alpha^i_j (u_j - l_j)| (\alpha^i \cdot u - \alpha^i \cdot l).
\end{aligned}$$

Thus, we derive that $\lim_{\epsilon \to 0^+} \|h_\epsilon(g) - \lfloor g \rfloor\|_1 = 0$. □

Lemma 3 implies that for any piecewise affine function g, we can approximate the value of the round-down of g over a bounded box domain using the continuous function h_ϵ.

2.3 Chvátal-Gomory Dual Function

For any fixed $b \in \mathcal{D}$, there exists a finite set of CG inequalities multipliers $\{u^{b,i}\}_{i=1}^r$ that describes the convex hull of the integral solution of $IP(b)$ [44]. We use the superscript b to emphasize that these CG multipliers are derived from the IP with right-hand side vector b. In addition, by denoting $b^1 := b$, $b^{i+1} := [b^i, \lfloor u^i b^i \rfloor]^T$, and $A^1 := A$, $A^{i+1} := [A^i, \lfloor u^i A^i \rfloor]^T$, we have that the optimal value of $\max\{c^T x | Ax \le b, x \in \mathbb{Z}_+^n\}$ is equal to

$$\begin{aligned} \max\ & c^T x \\ \text{subject to}\ & Ax \le b \\ & \lfloor u^{b,1} A^1 \rfloor x \le \lfloor u^{b,1} b^1 \rfloor \\ & \lfloor u^{b,2} A^2 \rfloor x \le \lfloor u^{b,2} b^2 \rfloor \\ & \qquad \vdots \\ & \lfloor u^{b,r} A^r \rfloor x \le \lfloor u^{b,r} b^r \rfloor \\ & x \ge 0. \end{aligned} \tag{3}$$

Assumption (Minimal CG Inequalities). In Equation (3), we assume that there are no redundant CG inequalities in solving the problem $IP(b)$, i.e., every face defined by a CG inequality is maximal and contains an optimal solution.

Since (3) is an LP, by strong LP duality, we also derive that the optimal value of (3) is the same as the following LP dual problem:

$$\begin{aligned} \min\ & p^T b + q_1^T \lfloor u^{b,1} b^1 \rfloor + \cdots + q_r^T \lfloor u^{b,r} b^r \rfloor \\ \text{subject to}\ & p^T A + q_1^T \lfloor u^{b,1} A^1 \rfloor + \cdots + q_r^T \lfloor u^{b,r} A^r \rfloor \ge c \\ & p, q_1, \ldots, q_r \ge 0. \end{aligned} \tag{4}$$

Since $b \in \mathcal{D}$, by its definition, the problem $IP(b)$ must have a finite optimal solution. By our assumption on the minimal CG inequalities, (3) has a finite optimal solution, and thus (4) also has a finite optimal solution. Let $p^b, q_1^b, \ldots, q_r^b$ be an optimal solution of (4), and consider the following function:

Definition 6. *We say that $f_b(\cdot) : \mathcal{D} \to \mathbb{R}$ is a* ***CG dual function*** *with respect to the right-hand side vector $b \in \mathcal{D}$ if for every $\beta \in \mathcal{D}$,*

$$f_b(\beta) = (p^b)^T \beta + (q_1^b)^T \lfloor u^{b,1} \beta^1 \rfloor + \cdots + (q_r^b)^T \lfloor u^{b,r} \beta^r \rfloor, \tag{5}$$

with $\beta^1 = \beta$, $\beta^{i+1} = [\beta^i, \lfloor u^i \beta^i \rfloor]^T$, where u^b, p^b and q^b are the CG multipliers described in (3) and optimal solution of (4) for right-hand side b, respectively.

Certainly, for a given right-hand side b, there is more than one set of CG inequalities that lead to the solution of $IP(b)$. Hence, there can be multiple CG dual functions associated with a right-hand side b. Since, by construction, a CG dual function is a CG function, and later in this section, we show that this class function plays an essential role in deriving a novel representation theorem for the IP value function, we want to dedicate a portion of this section to study the properties of CG dual functions. We first start with a simple observation for a CG dual function.

Proposition 1. *Given a CG dual function $f_b(\cdot) : \mathbb{R}_+^m \to \mathbb{R}_+$ where $b \in \mathcal{D}$, we have*

$$f_b(b) = z(b).$$

Proof. This comes directly from how we construct the function f_b. Since $z(b)$ is the optimal value of the IP $\max\{c^Tx|Ax \leq b, x \in \mathbb{Z}^n_+\}$ and $f_b(b)$ gives the objective value of (4), by LP strong duality, we must have $f_b(b) = z(b)$. $\square$

Proposition 1 says that the CG dual function gives the optimal objective at the corresponding right-hand side. This is of limited practical use because according to the definition of a CG dual function, we are obliged to incorporate all the necessary CG inequalities, thus constructing the CG dual function is as hard as solving the IP itself. Another way to view the CG dual functions is that they incorporate cutting plane information. This is a starting intuition for us to derive more interesting properties for this class of functions.

Proposition 2. *Consider a CG dual function $f_b(\cdot) : \mathbb{R}^m_+ \to \mathbb{R}_+$. If $LP(b)$ has an unique solution and $z(b) = z_{LP}(b)$, we have*

$$f_b(tb) = z_{LP}(tb), \text{ for every } t \in \mathbb{R}_+.$$

Proof. We have

$$f_b(b) = z(b) = z_{LP}(b).$$

By our assumption on the minimal set of added CG inequalities and the uniqueness of the solution of $LP(b)$, there is a vector p^b such that

$$f_b(\beta) = (p^b)^T\beta.$$

Thus, we have that $tf_b(tb) = tf_b(b) = tz_{LP}(b) = z_{LP}(tb) \ \forall\ t \in \mathbb{R}_+$. $\square$

Proposition 2 implies that for a right-hand side b where the $LP(b)$ and $IP(b)$ share the same solution, the CG dual function f_b is a linear function. It is because, in this case, p^b is the optimal LP dual extreme point.

Proposition 3. *For a fixed $b \in \mathcal{D}$, its corresponding CG dual function f_b is a feasible solution to the superadditive dual of the IP $\max\{cx|Ax \leq b, x \in \mathbb{Z}^n_+\}$, i.e., f_b is a feasible solution of*

$$\begin{aligned} \min\ & f(b) \\ s.t\ & f(a^j) \geq c_j \ \forall j \in [\![n]\!], \\ & f(0) = 0, \\ & f \text{ is non-decreasing and superadditive.} \end{aligned} \tag{6}$$

Proof. Since the floor function is superadditive, the CG multipliers $u^{b,i}$ for $i \in [\![r]\!]$ along with the dual variable p^b and q_i^b for $i \in [\![r]\!]$ are non-negative, the CG dual function corresponding with the right-hand side b is non-decreasing and superadditive. Moreover, by definition, we have $f_b(0) = 0$. In addition, from the constraint of (4), we have

$$f_b(a^j) \geq c_j \ \forall j \in [\![n]\!].$$

Hence, the CG dual function f_b is a feasible solution to the superadditive dual (6). $\square$

In a special case where the entries of the matrix A are non-negative, the dual problem can be written as an LP where each variable can be interpreted as an upper bound of the optimal value of the IP when the right-hand side vector is a certain integral vector. In particular, the following LP is equivalent to the superadditive dual [44]:

$$\begin{aligned} \min\ & F(b) \\ \text{s.t}\ & F(a^j) \geq c_j \ \forall j \in [\![n]\!], \\ & F(d_1) + F(d_2) - F(d_1 + d_2) \leq 0 \ \forall d_1, d_2, d_1 + d_2 \in D(b), \\ & F(0) = 0, F(d) \geq 0 \ \forall d \in D(b), \end{aligned} \tag{SDLP}$$

where $D(b) \coloneqq \{d \in \mathcal{D}|d \leq b\}$ and F is a $|D(b)|$-dimensional vector. By the feasibility of f_b from Proposition 3, we have the following immediate connection between f_b and the LP (SDLP).

Corollary 1. *Consider a CG dual function f_b and let the vector F_b be such that $F_b(d) = f_b(d)$ for every $d \in D(b)$. Then F_b is a solution of (SDLP).*

By Proposition 3, a CG dual function f_b is a feasible solution of (6), it is an upper bound on the IP value function z, i.e., $f_b(\beta) \geq z(\beta)$ for every $\beta \in \mathcal{B}$. Based on this property, we have the following observation between CG dual functions and the IP value function z.

Proposition 4. *Let $z(\beta)$ be the optimal value of $\max\{c^T x | Ax \leq \beta, x \in \mathbb{Z}^n_+\}$. Then*

$$z(\beta) = \min_{b \in \mathcal{D}} \{f_b(\beta)\} \ \forall \beta \in \mathcal{D}.$$

Proof. By Proposition 3, we have $f_b(\beta) \geq z(\beta)$ for every $b, \beta \in \mathcal{D}$, thus $\min_{b\in\mathcal{D}} f_b \geq z$. Moreover, by Proposition 1, we have $f_\beta(\beta) = z(\beta)$. Therefore, $z(\beta) = \min_{\beta\in\mathcal{D}} f_b(\beta) \ \forall \beta \in \mathcal{D}$. □

Certainly, it is not practical to solve an IP for every non-negative integral right-hand side. The only important meaning that it conveys is taking a minimum of multiple CG dual functions can give a better approximation of the value function z. However, we might not need an infinite number of CG dual functions to construct the IP value function z, as observed in the following examples.

Example 1. *: Consider the following integer knapsack (Figure 1):*

$$\begin{aligned} z(b) = \max\ & 3x_1 + x_2 \\ s.t\ & 2x_1 + x_2 \leq b \\ & x_1, x_2 \in \mathbb{Z}_+. \end{aligned} \tag{7}$$

Let $b = 1$ and solve the corresponding problem by adding a CG inequality with the multiplier $u = 1/2$. The integer knapsack is now equivalent to

$$\begin{aligned} z(b) = \max\ & 3x_1 + x_2 \\ s.t\ & 2x_1 + x_2 \leq 1 \\ & x_1 \leq 0 \\ & x_1, x_2 \geq 0. \end{aligned} \tag{8}$$

By solving the dual of (8), we construct the corresponding CG dual function f_1 as

$$f_1(\beta) = \lfloor \beta + \lfloor \frac{\beta}{2} \rfloor \rfloor.$$

The plot of f_1 is given Figure 1. Note that, in the definition of f_1, we use one additional outer round-down operation as compared to our definition of CG dual functions. This is because, in the definition of CG dual functions, we assume every right-hand side vector b is integral, and the outer round-down operator in this example is only used to cover the cases where b is not an integer. Interestingly, we can observe that the function f_1, constructed only by one right-hand side vector, is indeed the IP value function $z(b)$.

Example 2. *: Consider the following IP (Figure 2)*

$$\begin{aligned} z(b) = \max\ & x_1 + x_2 \\ s.t\ & 2x_1 + x_2 \leq b_1 \\ & x_1 + 2x_2 \leq b_2 \\ & x_1, x_2 \in \mathbb{Z}_+. \end{aligned} \tag{9}$$

We construct two CG dual functions, one corresponds with $b = [2, 0]^T$ and one corresponds with $b = [0, 2]^T$. The plot of the first function is in the leftmost, and the plot of the second function is in the middle of Figure

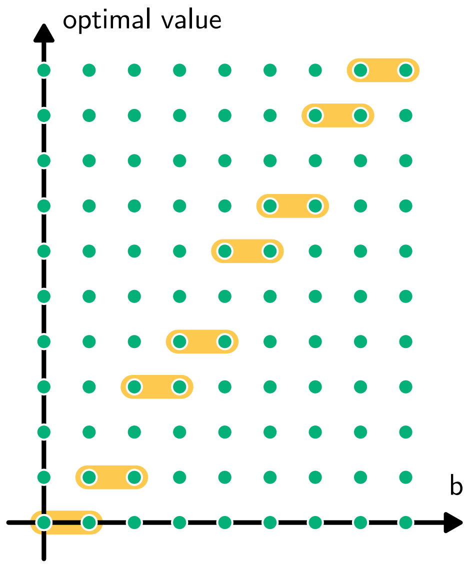

Figure 1: An illustration of a CG dual function in 1-dimensional space.

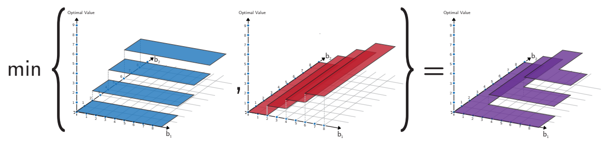

Figure 2: An illustration of the CG dual functions in 2-dimensional space.

2. When taking the minimum of these two functions, we obtain the IP value function z in the rightmost.

As observed by these two examples, when constructing the IP value function, we might only need to take the minimum of a finite number of CG dual functions. Next, we prove that this conjecture is indeed true for every IP.

2.4 Constructing The Integer Programming Value Function

In this subsection, we derive a "pattern" property of an IP value function. Generally speaking, we show that for any value function z, there exists a bounded domain $\mathcal{B}$ such that for any point β outside the domain, we can compute $z(\beta)$ based on the value of z over $\mathcal{B}$. The pattern of IP value functions was discussed in [19, Theorem 1]. However, in this theorem, the pattern property is only concerned with right-hand side vectors that are far away from the boundaries of some cones. Later, Alfant et al. [1, Proposition 3.2] also came up with a way to compute MIP (and thus applicable to IP) value functions given known values of the function at some points $\hat{\beta} < \beta$ using the optimal solutions of $IP(\beta)$. However, their result does not guarantee that the value function at every point in $\mathcal{D}$ can be computed using this property.

Let $\mathcal{S} := \{(c_1, a^1), \dots, (c_n, a^n), (0, e_1), \dots, (0, e_m), (-1, 0^m)\}$ and $\mathcal{C} := \text{cone}(\mathcal{S})$ denote the polyhedral cone generated by $\mathcal{S}$; see Figure 3 for an illustration. Let $\mathcal{F}$ be the (finite) set of facets of the polyhedral cone C. For each face $F \in \mathcal{F}$, we have that the set of m extreme rays defining F is a subset of $\mathcal{S}$ [24, Lemma 3]. We note that $\mathcal{F}$ can contain at most m facets whose extreme rays are $(-1, 0^m)$ and a set of $m-1$ unit vectors in the m-dimensional space. We denote the set of facets of $\mathcal{C}$ that excludes the facet containing $(-1, 0^m)$ by $\mathcal{F}'$.

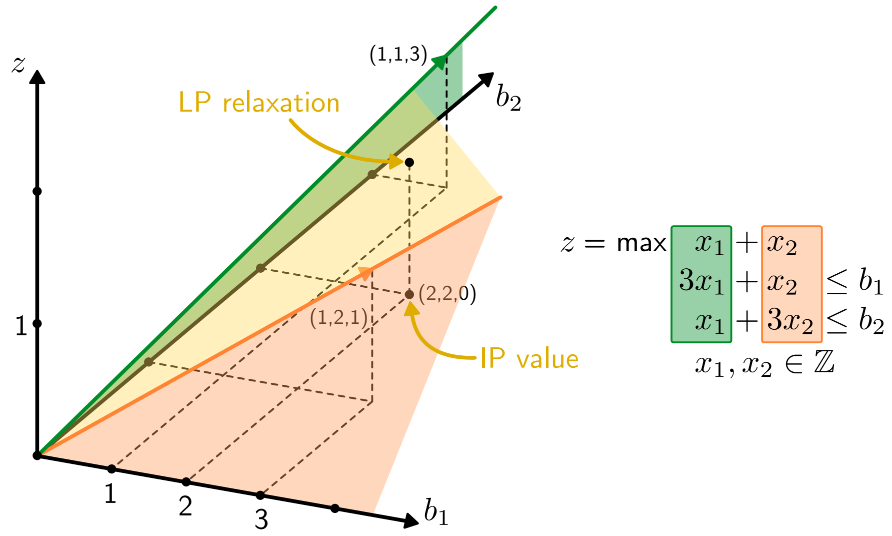

Figure 3: An illustration of cone $\mathcal{C}$. We only show here the 3 relevant facets of $\mathcal{C}$, while omitting the other 2 facets which contain the extreme rays $(-1, 0, 0)^T$. Intuitively, for a given right-hand side $b = (b_1, b_2)$, we move upward as far as possible until we reach a point belonging to one of the faces of cone $\mathcal{C}$, and the z-coordinate of which is the optimal value of the LP-relaxation. The optimal value of $z(b)$ is some point below the optimal value of the LP relaxation.

Lemma 4. *Let $F \in \mathcal{F}'$ be a facet of $\mathcal{C}$, and let $\{(\gamma_1^F, v_1^F), \dots (\gamma_m^F, v_m^F)\} \subset \mathcal{S}$ be the set of finite extreme rays defining F. Then, for every non-negative integer k_i for $i \in [\![m]\!]$ such that $IP(k_1 v_1^F + \cdots + k_m v_m^F)$ has a finite optimal solution, we have:*

$$z(k_1 v_1^F + \cdots + k_m v_m^F) = k_1 \gamma_1^F + \cdots + k_m \gamma_m^F.$$

Proof. Since $F \in \mathcal{F}'$, every v_i^F is either a column of A or a unit vector, and every γ_i^F is a component of the objective c or is equal to 0. We have that $z(k_1 v_1^F + \cdots + k_m v_m^F) \ge k_1 \gamma_1^F + \cdots + k_m \gamma_m^F$, as we can easily find a feasible solution of $z(k_1 v_1^F + \cdots + k_m v_m^F)$ whose objective is equal to $k_1 \gamma_1^F + \cdots + k_m \gamma_m^F$ by setting the value of the variable corresponds to the column v_j^F equal to k_j. By contradiction, suppose that

$$z(k_1 v_1^F + \cdots + k_m v_m^F) - (k_1 \gamma_1^F + \cdots + k_m \gamma_m^F) > 0.$$

Let $z_{LP}(b)$ denote the LP relaxation value function of $z(b)$. Take

$$\epsilon := z_{LP}(k_1 v_1^F + \dots k_m v_m^F) - (k_1 \gamma_1^F + \cdots + k_m \gamma_m^F) > 0.$$

Let x^* be the optimal primal solution of $LP(k_1 v_1^F + \cdots + k_m v_m^F)$; we have

$$\begin{cases} c^T x^* - \epsilon = k_1 \gamma_1^F + \cdots + k_m \gamma_m^F \\ Ax^* + I_m s^* = k_1 v_1^F + \cdots + k_m v_m^F, \end{cases}$$

where $s^* := \sum_{i=1}^m k_i v_i^F - Ax^* \ge 0$. Hence we have that

$$\begin{bmatrix} c \\ A \end{bmatrix} x^* + \epsilon \begin{bmatrix} -1 \\ 0^m \end{bmatrix} + \begin{bmatrix} 0 \\ I_m \end{bmatrix} s^* = k_1 \begin{bmatrix} \gamma_1^F \\ v_1^F \end{bmatrix} + \cdots + k_m \begin{bmatrix} \gamma_m^F \\ v_m^F \end{bmatrix}. \tag{10}$$

The right-hand side of (10) is a vector belonging to the facet F, while the left-hand side is a vector that does not belong to F as $\epsilon > 0$. Hence

$$z(k_1 v_1^F + \dots k_m v_m^F) = k_1 \gamma_1^F + \cdots + k_m \gamma_m^F.$$

□

For a set of vectors V of $\mathcal{C}$, we denote its projection onto the space of the right-hand side b, which is $\mathbb{R}^m$, by

$$\text{Proj}_{\mathbb{R}^m}(V) := \{v \in \mathbb{R}^m | \exists\ \gamma \in \mathbb{R}_+ \text{ s.t } (\gamma, v) \in V\}.$$

Lemma 5. *Let $F := cone((\gamma_1, v_1), \dots, (\gamma_m, v_m)) \in \mathcal{F}'$ be a facet of C and $\beta \in Proj_{\mathbb{R}^m}(F)$ be represented as*

$$\beta = k_1 v_1 + \cdots + k_m v_m + \bar{\beta},$$

where k_i are non-negative integers and $\bar{b} < v_j\ \forall j \in [\![m]\!]$. Then, there exists non-negative integers $K_1, \dots, K_m$ such that for any $j \in [\![m]\!]$ satisfying $k_j > K_j$ and $\gamma_i > 0$, we have $z(\beta) = z(\beta - v_j) + \gamma_j$.

Proof. We only prove the existence of K_1; the existence of $K_2, \dots, K_m$ can be proved similarly. We have that $z_{LP}(\beta) - z(\beta) \leq M_{A,c}$ for every $\beta \in \mathcal{D}$, where the constant $M_{A,c}$ only depends on the constraint matrix A and the objective coefficient c [16, Corollary 2]. Therefore, we have that

$$\begin{aligned} z(\beta + v_1) - z(\beta) &\leq z_{LP}(\beta + v_1) - z(\beta) && \text{(11a)} \\ &\leq z_{LP}(\beta) + z_{LP}(v_1) - z(\beta) && \text{(11b)} \\ &= z_{LP}(\beta) - z(\beta) + \gamma_1 && \text{(11c)} \\ &\leq M_{A,c} + \gamma_1. && \text{(11d)} \end{aligned}$$

Inequality (11a) follows from the fact that $z \leq z_{LP}$. (11b) is based on the piecewise linear property of an LP value function. Finally, $z_{LP}(v_1) = \gamma_1$ because F is a facet of C.

Let x_{j^*} be the variable corresponding to the column of A that corresponds to v_1. Let $f_0(\beta)$ be an optimal value of the LP which is obtained by the relaxation of $z(\beta)$ with an additional constraint $x_{j^*} = 0$, i.e.,

$$\begin{aligned} f_0(\beta) := \max\ & c^T x \\ & Ax \leq \beta \\ & x_{j^*} = 0 \\ & x \geq 0. \end{aligned} \tag{12}$$

We have that $z_{LP}(v_1) > f_0(v_1)$ because (γ_1, v_1) is an extreme ray of C. Let $\epsilon := z_{LP}(v_1) - f_0(v_1)$ and $K_1 := \lceil \frac{M_{A,c}+\gamma_1}{\epsilon} \rceil$. We will prove that if $\beta = k_1 v_1 + \cdots + k_m v_m + \bar{\beta}$ and $k_1 \geq K_1$, then $z(\beta) = z(\beta - v_1) + \gamma_1$. To do so, we show that there exists an optimal solution $\bar{x}$ of $z(\beta)$ for which $\bar{x}_{j^*} \geq 1$. By contradiction, suppose that $\bar{x}_{j^*} = 0$ in every optimal solution of $IP(\beta)$, We have

$$\begin{aligned} z(\beta) \leq f_0(\beta) &= f_0(k_1 v_1 + \bar{\beta}) + f_0(k_2 v_2 + \dots k_m v_m) \\ &= f_0(k_1 v_1 + \bar{\beta}) + k_2 \gamma_2 + \dots k_m \gamma_m. \end{aligned}$$

On the other hand, we also have

$$\begin{aligned} z(\beta + v_1 - \bar{\beta}) &= z((k_1 + 1) v_1 + \cdots + k_m v_m) \\ &= (k_1 + 1)\gamma_1 + \cdots + k_m \gamma_m \text{ (by Proposition 2)}. \end{aligned}$$

Hence, we have

$$\begin{aligned} z(\beta + v_1 - \bar{\beta}) - z(\beta) &\geq (k_1 + 1)\gamma_1 - f_0(k_1 v_1 + \bar{\beta}) \\ &\geq (k_1 + 1)\gamma_1 - f_0((k_1 + 1) v_1) \\ &= (k_1 + 1)(\gamma_1 - f_0(v_1)) \\ &> \frac{M_{A,c} + \gamma_1}{\epsilon} \times \epsilon = M_{A,c} + \gamma_1, \end{aligned}$$

which contradicts Equation (11) as $z(\beta + v_1 - \bar{\beta}) \leq z(\beta + v_1)$. Hence, there exists a solution of $z(\beta)$ where $x_{j^*} \geq 1$. Thus,

$$z(\beta) = z(\beta - v_1) + \gamma_1\ \forall k_1 > K_1.$$

□

Suppose that for every face $F \in \mathcal{F}'$, we have a set of non-negative integers $\{K_1^F, \dots, K_m^F\}$ corresponding to each extreme ray of F. Let $\mathcal{K} := \max_{F\in\mathcal{F}', i\in[\![m]\!]} \|K_i^F v_i^F\|_1$. Since, for every vector $\beta \in \mathcal{D}$ where $z(\beta)$ is feasible, there exists a face $F \in \mathcal{F}'$ such that $(z_{LP}(\beta), \beta) \in F$. If $\|\beta\|_1 \geq \mathcal{K}$ and β is written in the form of $\beta = k_1^F v_1^F + \dots + k_m^F v_m^F + \bar{\beta}$, where $\bar{\beta} \leq v_i^F$ for every $i \in [\![m]\!]$, there must exist $i \in [\![m]\!]$ such that $k_i^F > K_i^F$. Thus by Lemma 5, we have $z(\beta) = z(\beta - (k_i^F - K_i^F)v_i^F) + (k_i^F - K_i^F)\gamma_i^F$. Given a vector $\beta \in \mathbb{R}^m_+$, to find a face $F \in \mathcal{F}'$, and represent as $b = k_1^F v_1^F + \dots + k_m^F v_m^F + \bar{b}$, we simply need to solve the relaxation $LP(\beta)$.

We can interpret Lemma 5 as describing the pattern of the IP value function, i.e., when we have enough point evaluation of z, we can use these values to determine the value of z at unevaluated points β without solving an IP. On the other hand, as discussed earlier, for a fixed right-hand side vector β, there exists a set of CG multipliers that gives the convex hull of the feasible domain of $z(\beta)$. This observation and the pattern property from Lemma 5 raises a natural question of whether we can reuse the same CG multiplier for $z(\beta')$ for a different right-hand side $\beta' \neq \beta$.

Lemma 6. *There exists a finite set $\mathcal{L}$ such that the function $z_l : \mathcal{D} \to \mathbb{R}$ defined as $z_l(\beta) := min_{b\in\mathcal{B}}\{f_b(\beta)\}$ is a lower bound of the LP value function z_{LP}, i.e.,*

$$z_l(\beta) \leq z_{LP}(\beta) \ \forall \beta \in \mathcal{D}.$$

Proof. Since z_{LP}, the value function of the LP relaxation is a concave function that is obtained by taking the minimum of a finite set of linear functions, we denote

$$z_{LP}(\beta) = \min_{i\in[\![|\mathcal{P}|]\!]} (p^i)^T \beta,$$

where $\mathcal{P}$ denotes the set of extreme points of $\{p \in \mathbb{R}^m | p^T A \geq c, p \geq 0\}$. For every extreme point p^i in $\mathcal{P}$, let b^i denote the right-hand side vector for which p^i is the unique optimal solution to the dual of $LP(b^i)$. Furthermore, for every $i \in [\![|\mathcal{P}|]\!]$, let $\bar{A}^i$ be an optimal basis of $LP(b^i)$. By strong duality, we have $z_{LP}(|\det(\bar{A}^i)|b^i) = (p^i)^T |\det(\bar{A}^i)| b^i$. Moreover, since the optimal basic variables of $LP(|\det(\bar{A}^i)|b^i)$ are $(\bar{A}^i)^{-1}|\det(\bar{A}^i)|\bar{b}^i$ where $\bar{b}^i$ is sub-vector of b^i corresponding to the basis $\bar{A}^i$, we have that the solution are integral. Thus, we derive that $z_{LP}(|\det(\bar{A}^i)|b^i) = z(|\det(\bar{A}^i)|b^i)$. Therefore, by letting $\mathcal{L} = \{|\det(\bar{A}^i)|b^i | \forall i \in [\![|\mathcal{P}|]\!]\}$, we have $z_l(\beta) \leq z_{LP}(\beta)$ for every $\beta \in \mathbb{R}^m$. ☐

Now, we use Lemma 5 and Lemma 6 to derive the main theorem of this section. We construct a function that is "sandwiched" between the LP relaxation z_{LP} from above and the IP value function z from below. Then, we rely on the property that the function z behaves in a pattern as described in Lemma 5: when our function agrees with z for enough number points, it must agree with z everywhere else.

Theorem 1. *There exists a finite set $\mathcal{B} \subset \mathcal{D}$ such that*

$$z(\beta) = \min_{b\in\mathcal{B}}\{f_b(\beta)\} \ \forall \beta \in \mathcal{D}.$$

Proof. For every facet F of C, let $b^F := K_1^F v_1^F + \dots K_m^F v_1^F$, where K_i^F and v_i^F are defined in Lemma 5, and let $\mathcal{L}$ be the set of vectors that satisfies the condition stated in Lemma 6. Given the vectors b^F for every facet of F of C and $\mathcal{L}$, we choose $\mathcal{B} = \{b \in \mathcal{D} | b \in \mathcal{L} \text{ or } \exists F \in \text{facet}(C) \text{ s.t } b \leq b^F\}$. We will show that the function $f^*(\beta) := \min_{b\in\mathcal{B}}\{f_b(\beta)\}$ equals the IP value function z at every integral point by contradiction.

By definition, we must have $f^*(\beta) = z(\beta)$ for every $b \in \mathcal{B}$ and $f^*(\beta) \geq z(\beta)$ for every $b \in \mathcal{D}$ since $f^*(\beta)$ is a feasible solution to the superadditive dual. By contradiction, suppose that there exists $\bar{\beta} \notin \mathcal{B}$ and $\epsilon := f^*(\bar{\beta}) - z(\bar{\beta}) > 0$. By Lemma 5, since $\bar{\beta}$ is outside $\mathcal{B}$, we can decompose $\bar{\beta}$ into sum of β_1 and β_2, where

$\beta_1 \in \mathcal{B}$, while $\beta_2 = \sum_{i=1}^m k_i v_i^F$ for some facet F of C.

$$\begin{aligned} & f^*(\bar{\beta}) - z(\bar{\beta}) > 0 \\ \Leftrightarrow \quad & f^*(\beta_1+\beta_2) - z(\beta_1+\beta_2) > 0 \\ \Leftrightarrow \; & f^*(\beta_1+\beta_2) - z(\beta_1) + z(\beta_2) > 0 \\ \Leftrightarrow \quad & f^*(\beta_1+\beta_2) - f^*(\beta_1) = z_{LP}(\beta_2) + \epsilon. \end{aligned} \tag{13}$$

Consider the CG functions $g^* : \mathbb{R} \to \mathbb{R}$ defined as $g^*(t) = f(\beta_1 + t\beta_2)$ for $t \in \mathbb{Z}$. Let c^* be the carrier of g^* [11, Definition 2.9]. Since g^* is univariate, $g^*(0) = f^*(\beta_1)$, and g^* is non-decreasing, there must exist $\alpha \geq 0$ such that $c^*(t) = \alpha t + f^*(\beta_1)$. We have $0 \leq c^*(t) - g^*(t) \leq r^*$, where r^* is the CG rank of g^*, and $\gamma^*(t) \coloneqq c^*(t) - g^*(t)$ is periodic. Let T denote the periodicity of γ^*, we have

$$c^*(1+T) + T\gamma^*(1) + Tf^*(\beta_1) \geq (1+T)c^*(1) \tag{14a}$$
$$\Leftrightarrow \; c^*(1+T) - \gamma^*(1+T) + Tf^*(\beta_1) \geq (1+T)(c^*(1) - \gamma^*(1)) \tag{14b}$$
$$\Leftrightarrow \; f^*(\beta_1 + T\beta_2) + Tf^*(\beta_1) \geq (1+T)f^*(\beta_1+\beta_2) \tag{14c}$$
$$\Leftrightarrow \; f^*(\beta_1 + (1+T)\beta_2) - f(\beta_1) \geq (1+T)(f^*(\beta_1+\beta_2) - f^*(\beta_1)). \tag{14d}$$

We have (14a) based on the linearity of c^* and the fact that γ^* is always non-negative. To derive (14b), we subtract $(1+T)\gamma^*(1)$ from both sides and apply the periodic property. Intuitively, (14d) means that if we increase the input of f^* by $T\beta_2$, the increase in f^* will increase at least linearly. In combination with (13), we have

$$\begin{aligned} f^*(\beta + (1+\tau T)\beta_2) &\geq (1+\tau T)(f^*(\beta_1+\beta_2) - f^*(\beta_1)) + f^*(\beta_1) \\ &\geq (1+\tau T)(z_{LP}(\beta_2)+\epsilon) + f^*(\beta_1). \end{aligned}$$

However, this mean that, as $\tau \to +\infty$, because $\epsilon > 0$, $f^*(\beta_1+(1+\tau T)\beta_2)$ will grow larger than the LP relaxation value $z_{LP}(\beta_1 + (1+\tau T)\beta_2)$, which contradicts our choice of $\mathcal{L}$. Therefore, we have $z(\beta) = f^*(\beta)\ \forall \beta \in \mathcal{D}$. □

Corollary 2. *For every IP value function z, there exists a CG function f whose carrier's coefficients are non-negative such that $z = f$.*

Proof. Since each f_b in Theorem 1 is a CG dual function, its coefficients are non-negative. From Theorem 1, every IP value function can be written as a minimization of a finite number of CG dual functions. Therefore, the coefficients of its carrier are non-negative. □

3 Tree Representation Theorem of the IP Value Function

In this section, we derive a NN architecture that can approximate an IP value function. We respectively denote the three CG operators as:

$$\begin{aligned} \text{linear operator}: &\ \Lambda^{\mu_1,\mu_2}(f_1,f_2)(v) = \mu_1 f(v) + \mu_2 f(v), \\ \text{round-down operator}: &\ \lfloor f \rfloor(v) = \lfloor f(v) \rfloor, \\ \text{minimum operator}: &\ \min(f_1,f_2)(v) = \min\{f_1(v), f_2(v)\}. \end{aligned}$$

In addition, we also denote $\mathcal{H} \coloneqq \{\Lambda^{\alpha,\beta}(\cdot,\cdot) | \mu_1,\mu_2 \in \mathbb{R}_+\} \cup \{\lfloor\cdot\rfloor, \min\{\cdot,\cdot\}\}$ as the set of all CG operators on a function in m-dimensional space. Finally, we use $\mathcal{G}$ to denote the class of CG functions whose input is m-dimensional, and $\mathcal{L}_\emptyset$ as the class of m-dimensional linear functions, i.e.,

$$\mathcal{L}_\emptyset \coloneqq \{f | \exists \lambda \in \mathbb{R}^m \text{ s.t } f(v) = \lambda^T v\ \forall v \in \mathbb{R}^m\}.$$

For a class of functions F, we define F_h to be the class of functions in F equipped with a operator $h \in \mathcal{H}$ to be

$$F_h := \begin{cases} F \cup \{h(f_1, f_2) | f_1, f_2 \in F\}, \text{ if } h \in \{\Lambda^{\mu_1,\mu_2}, \min\}, \\ F \cup \{\lfloor f \rfloor | f \in F\}, \text{ if } h = \lfloor \cdot \rfloor. \end{cases} \tag{15}$$

A class of functions can also be stacked with multiple CG operators. We define, inductively $F_{h_1,\ldots,h_r}$ to be the class of functions $F_{h_1,\ldots,h_{r-1}}$ equipped with the CG operator h_r. Using this notation, we can derive a simple representation for the class of rank r CG functions.

Proposition 5. *Let $\mathcal{G}_r$ be the class of CG functions of rank at most r. We have*

$$\mathcal{G}_r = \begin{cases} \emptyset, \text{ if } r = 0, \\ \cup_{(h_1,\ldots,h_r) \subseteq \mathcal{H}^r} \mathcal{L}_{h_1,\ldots,h_r} \text{ otherwise,} \end{cases} \tag{16}$$

where $\mathcal{H}^r$ denotes the r^{th} Cartesian product for the set of CG operators $\mathcal{H}$.

Proof. If $r = 0$, then $\mathcal{G}_0$ is the set of linear functions. Hence $\mathcal{G}_0 = \mathcal{L}_\emptyset$. By induction, suppose that the hypothesis is true for r; we prove it is also true for $r + 1$. By definition of CG functions, we have $\mathcal{L}_{h_1,\ldots,h_{r+1}} \subseteq \mathcal{G}_{r+1}$. Thus, we only need to show that $\mathcal{G}_{r+1} \subseteq \cup \mathcal{L}_{h_1,\ldots,h_{r+1}}$.

If a function $f \in \mathcal{G}_{r+1}$, then exactly one of the following must be true for the last CG operation of h:

1. $f = \lfloor f' \rfloor$ for some $f' \in \mathcal{G}_r$. By the induction hypothesis $f' \in \cup \mathcal{L}_{h_1,\ldots,h_r}$, and thus $f \in \cup \mathcal{L}_{h_1,\ldots,h_r,\lfloor \cdot \rfloor}$.
2. $f = \min\{f_1, f_2\}$ for some $f_1, f_2 \in \mathcal{G}_r$. By the induction hypothesis $f_1, f_2 \in \cup \mathcal{L}_{h_1,\ldots,h_r}$, thus $f \in \cup \mathcal{L}_{h_1,\ldots,h_r,\min}$.
3. $f = \mu_1 f_1 + \mu_2 f_2$ for some $\mu_1, \mu_2 \in \mathbb{Q}_+$ and $f_1, f_2 \in \mathcal{G}_r$. By the induction hypothesis $f_1, f_2 \in \cup \mathcal{L}_{h_1,\ldots,h_r}$, thus $f \in \cup \mathcal{L}_{h_1,\ldots,h_r,\Lambda^{\mu_1,\mu_2}}$.

Hence, $\mathcal{G}_{r+1} = \cup \mathcal{L}_{h_1,\ldots,h_r,h_{r+1}}$. □

Now, we put the operators of the CG function in the context of NNs. In a NN, a neuron, also known as a node or unit, is the fundamental computational unit. It receives one or multiple inputs, performs a weighted summation of these inputs, adds an optimal bias term, and then applies an activation function to produce an output (usually non-linear). A layer in a NN is a collection of neurons that process input data together. Layers are organized in a hierarchical manner, where each layer receives input from the preceding layer and sends its output to the subsequent layer [6]. There are three types of layers in a NN: *Input Layers*, *Hidden Layers*, and *Output Layers*. The input layer receives input data, which could be features from a dataset or, in our case, the right-hand side vector of the IPs. Hidden layers are intermediate layers between the input and output layers. The output layer produces the network's predictions or outputs based on the processed information from the hidden layers. In the context of IP value functions, the dimension of the output layer is one.

Theorem 2. *Given a real number $\delta > 0$, a bounded input domain $\mathbf{B} := \{b \in \mathcal{D} | \ \|b\|_1 \le \mathcal{K}\}$, and a CG function $z(b)$ of rank r with its carrier's coefficient bounded by a positive value $\mathcal{M}$, there exists a NN f with $O(r \log(m\mathcal{M}\mathcal{K}))$ layers and $O(2^{r+1} + r \log(m\mathcal{M}\mathcal{K}))$ neurons such that*

$$\int_{\mathbf{B}} |f(b) - z(b)| db < \delta.$$

Proof. The proof is based on the construction of the function z. Certainly, when z is an affine function, we can use a single neuron to model z as a NN exactly. Suppose the theorem is true for every CG function that uses r or fewer operations; we prove that it is also true for a CG function that contains $r+1$ operations.

Case 1: Suppose $z = \mu_1 z_1 + \mu_2 z_2$, where $\mu_1, \mu_2 \in \mathbb{R}_+$ and z_1, z_2 are CG functions of rank smaller or equal to r. By the induction hypothesis, there exist two NNs f_1 and f_2 such that

$$\int_{\mathbf{B}} |f_1 - z_1| \leq \frac{\delta}{2\mu_1} \text{ and } \int_{\mathbf{B}} |f_2 - z_2| \leq \frac{\delta}{2\mu_2}.$$

We construct a NN representing f which contains f_1, f_2, and a final layer with one neuron whose input is the output of f_1, f_2 and whose weight is (α, β) so that we have $f = \alpha f_1 + \beta f_2$. Hence,

$$\int_{\mathbf{B}} |f - z| \leq \mu_1 \int_{\mathbf{B}} |f_1 - z_1| + \mu_2 \int_{\mathbf{B}} |f_2 - z_2| \leq \delta.$$

Case 2: Suppose $z = \min\{z_1, z_2\}$. By the induction hypothesis, there exist two NNs f_1 and f_2 such that

$$\int_{\mathbf{B}} |f_1 - z_1| \leq \frac{\delta}{2} \text{ and } \int_{\mathbf{B}} |f_2 - z_2| \leq \frac{\delta}{2}.$$

We then construct a NN f, which contains f_1, f_2, and a final min layer. We have

$$\int_{\mathbf{B}} |f - z| = \int_{\mathbf{B}} |\min\{f_1, f_2\} - \min\{z_1, z_2\}| \leq \int_{\mathbf{B}} |f_1 - z_1| + \int_{\mathbf{B}} |f_2 - z_2| \leq \delta.$$

Case 3: Suppose $z = \lfloor z' \rfloor$. Suppose we have a NN f' that approximates z', and the NN f is constructed from f' with a final layer equal to h_ϵ. By Lemma 3, we choose ϵ so that $\|h_\epsilon(z') - \lfloor z' \rfloor\|_1 \leq \frac{\delta}{2}$. Furthermore, we choose the NN f' such that $\|z' - f'\|_1 \leq \frac{\delta}{2}$. We have

$$\begin{aligned}\int_{\mathbf{B}} |f - z| &= \int_{\mathbf{B}} |h_\epsilon(f') - h_\epsilon(z') + h_\epsilon(z') - \lfloor z' \rfloor| \\ &\leq \int_{\mathbf{B}} |h_\epsilon(f') - h_\epsilon(z')| + \int_{\mathbf{B}} |h_\epsilon(z') - \lfloor z' \rfloor| \\ &\leq \int_{\mathbf{B}} |f' - z'| + \int_{\mathbf{B}} |h_\epsilon(z') - \lfloor z' \rfloor|.\end{aligned} \tag{17}$$

Hence, we derive

$$\int_{\mathbf{B}} |f - z| \leq \frac{\delta}{2} + \frac{\delta}{2} = \delta.$$

The number of piece in h_ϵ is bounded by $m\mathcal{KM}$. By Montufar et al. [33, Corollary 5], there exists a NN with $O(\log(m\mathcal{KM})$ that represent h_ϵ. □

Constructing a NN based on Theorem 2 can be viewed as forming a balanced binary tree and then a fully connected layer connecting the input with the tree's leaves. For an illustration, see Figure 4. Knowing the order of the CG operations used in the construction of $z(b)$ allows us to assign each NN layer the corresponding operations.

4 Block Representation Theorem of the IP Value Function

In the previous section, we have shown the existence of a NN with a bounded size that approximates an IP value function to an arbitrarily small L_1 error. However, this NN architecture does not guarantee that the approximation is monotone or superadditive - both structural properties of an IP value function. In the next sections, based on the structural results of Section 2, we derive another representation theorem. Using this NN architecture, we can impose non-negativity constraints on the weights of the NN and replace ReLU with round-down activation to enforce monotone and superadditive properties, while not hurting the ability of

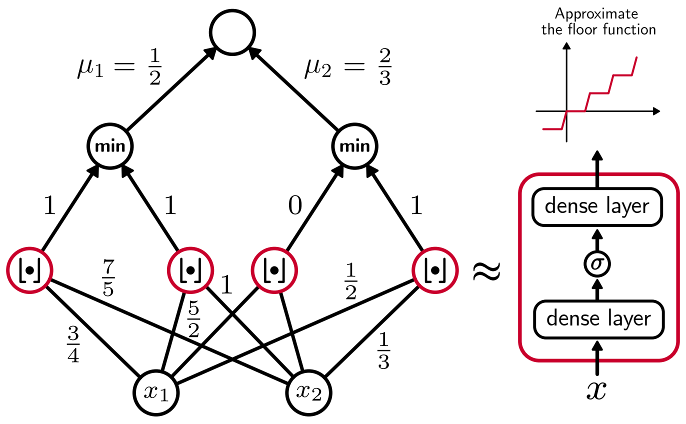

Figure 4: An example of the NN with exponential size width and linear size depth with respect to CG rank. Consider the function $f(x_1, x_2) = \frac{1}{2}\min\{\lfloor\frac{3}{4}x_1 + \frac{7}{5}x_2\rfloor, \lfloor\frac{5}{2}x_1 + x_2\rfloor\} + \frac{2}{3}\lfloor\frac{1}{2}x_1 + \frac{1}{3}x_2\rfloor$. We can see that $f(x_1, x_2) \in \mathcal{L}_{\lfloor\cdot\rfloor,\min,\Lambda^{1/2,2/3}}$. Based on the order of CG operations, we can construct a NN that models $f(x_1, x_2)$ top-down and in reverse order of the operation. In particular, starting from a single neuron corresponding to the output of the net, we create two children nodes, which will be the next layer. And we keep "branching" until we reach depth r, which is the rank of the CG function. The activation function for $\Lambda^{1/2,2/3}$ is linear and $\lfloor\cdot\rfloor$ is the round-down function, which can be approximated using a smaller network.

NNs to represent IP value function. In addition, the Block Representation Theorem allows a NN training framework that guarantees an upper approximation of the IP value function.

We now construct a NN that can approximate the IP value function $z(\beta)$ based on Theorem 1. Naturally, we want to have a structure that can represent a function of the form as in Equation (5). In Figure 5, we have a NN with $k+1$ hidden neurons, where exactly one of them has no activation function, while the remaining k has the floor operator as activation functions (or functions that approximate the floor function). Formally, we denote $z_0, z_1, \ldots z_k$ as outputs of the $k+1$ neurons. Then, the output of a CG block with respect to an input β is computed as follows:

$$\begin{aligned}\text{output} &= z_0 + q_1 z_1 + \cdots + q_k z_k \\ z_0 &= \beta \cdot p, \\ z_j &= \lfloor z_1 \bar{u}_1^j + \cdots + z_{i-1}\bar{u}_{j-1}^j + \beta \cdot \tilde{u}^j \rfloor \ \forall j \in [\![k]\!],\end{aligned} \tag{18}$$

where p, q, u are weights of the CG block. In particular, p is the weight of the neuron corresponding with z_0, and q is the weight of the output neuron. Since for each CG neuron, there are two types of weights - weights for the input and weights for the previous CG neurons - we use $\tilde{u}$ for the input weight and $\bar{u}$ to denote the previous CG neurons' weights. We use the term "CG neuron" to refer to a neuron that takes the right-hand side β and the outputs of all previous neurons as input. In addition, we call a NN consisting only of CG neurons a CG block. When a NN is constructed by taking the minimum of multiple CG blocks, we call it a CG Neural Network (CGNN).

Theorem 3. *Given an IP value function z, there exists a CGNN with a finite number of CG blocks each with a finite number of neurons that equals the IP value function $z(\beta)$.*

Proof. By Theorem 1, if each block is equal to a CG dual function $f_b(\beta)$ for every $b \in \mathcal{B}$, then the entire NN is exactly equal to the IP value function $z(\beta)$. We define $N := \|\mathcal{B}\|$. In addition, since every CG dual function requires a finite number of round-down operations, for each block $i \in [\![N]\!]$, we only need a finite number of k_i neurons to represent the CG dual function. $\square$

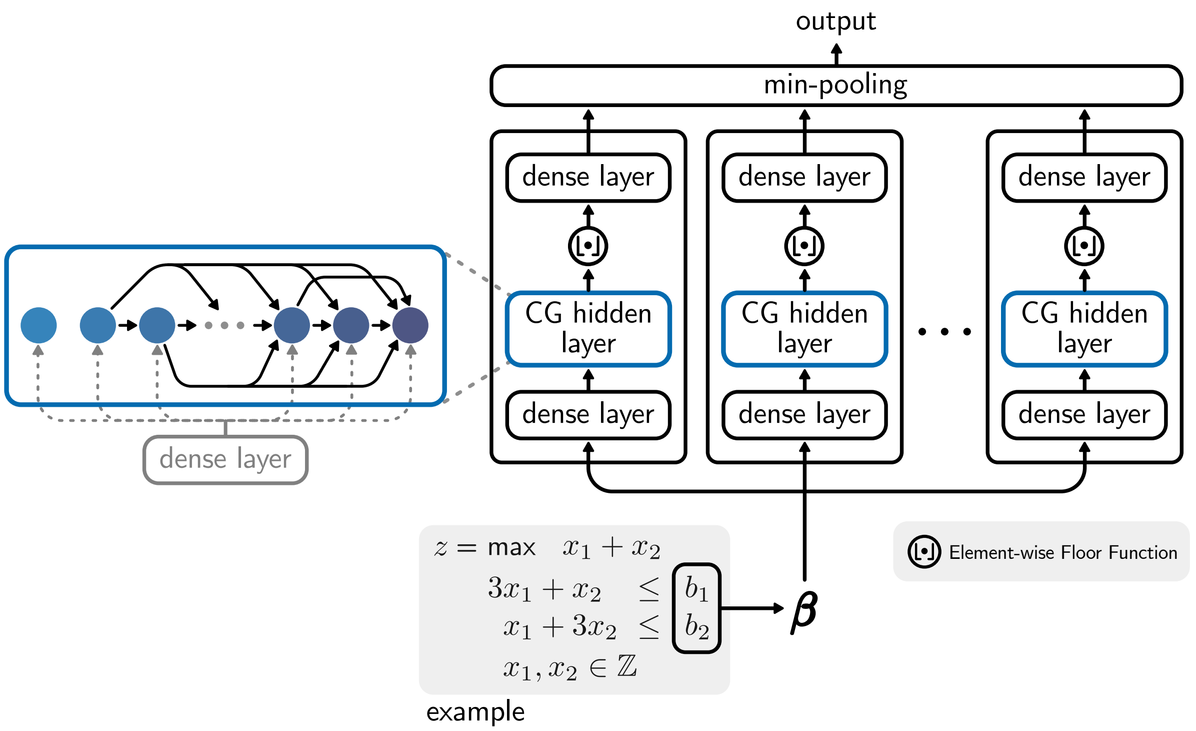

Figure 5: CGNN Architecture: (Left) An illustration of a CG block. (Right) An illustration of a CGNN containing multiple CG blocks and a Min-Pooling Layer.

4.1 Mixed-Integer Formulation for Constructing IP Value Functions

Now, we discuss an MIP formulation that guarantees a superadditive function that upper bounds the IP value function. Based on Theorem 3, an IP value function can be represented by a finite number of blocks, where each block is parameterized by a finite set of weights. We first derive the following bilinear integer formulation for the superadditive dual feasibility of one block with k CG neurons.

$$z^i_j = \lfloor z^i_1 \bar{u}^j_i + \cdots + z^i_{j-1}\bar{u}^j_{j-1} + a^i \cdot \tilde{u}^j \rfloor \ \forall i \in [\![n]\!], j \in [\![k]\!], \tag{19a}$$
$$a^i \cdot p + q_1 z^i_1 + \cdots + q_k z^i_k \ge c_i \ \forall i \in [\![n]\!], \tag{19b}$$
$$p, q_i, \bar{u}^j_i, \tilde{u}^j \ge 0, z^i_j \in \mathbb{Z} \ \forall i \in [\![n]\!], j \in [\![k]\!]. \tag{19c}$$

In (19), we use the variables z^i_j for the post-activation values of CG neuron j with respect to input a^i. In addition to the variables z, by introducing variables $p, q, \bar{u}, \tilde{u}$ for each block, where each block has k_r CG neurons, we can extend (19) for a superadditive dual feasible formulation to a CGNN with N blocks:

$$z^i_{j,r} = \lfloor z^i_{1,r} \bar{u}^j_{i,r} + \cdots + z^i_{j-1,r}\bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r \rfloor \ \forall i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!], \tag{20a}$$
$$a^i \cdot p_r + q_{r,1} z^i_{1,r} + \cdots + q_{r,k_r} z^i_{k_r,r} \ge c_i \ \forall i \in [\![n]\!], r \in [\![N]\!], \tag{20b}$$
$$p_r, q_r, \bar{u}^j_{i,r}, \tilde{u}^j_r \ge 0, \ \ z^i_{j,r} \in \mathbb{Z} \ \forall i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!]. \tag{20c}$$

Corollary 3. *Let $\mathcal{B}$ be the finite set described in Theorem 1 so that that $z(\beta) = \min_{b \in \mathcal{B}} f_b(\beta)$. Consider the following problem:*

$$
\begin{aligned}
&\min \sum_{b\in\mathcal{B}} w_b && \text{(21a)}\\
s.t\ &z^i_{j,r} = \lfloor z^i_{1,r}\bar{u}^j_{1,r} + \cdots + z^i_{j-1,r}\bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r \rfloor \ \forall i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!], && \text{(21b)}\\
&z^b_{j,r} = \lfloor z^b_{1,r}\bar{u}^j_{1,r} + \cdots + z^b_{j-1,r}\bar{u}^j_{j-1,r} + b \cdot \tilde{u}^j_r \rfloor \ \forall b \in [\![\mathcal{B}]\!], j \in [\![k_r]\!], r \in [\![N]\!], && \text{(21c)}\\
&a^i \cdot p_r + q_{r,1} z^1_{i,r} + \cdots + q_{r,k_r} z^{k_r}_{i,r} \geq c_i \ \forall i \in [\![n]\!], r \in [\![N]\!], && \text{(21d)}\\
&\min_{r\in N}\{b \cdot p_r + q_{r,1} z^b_{1,r} + \cdots + q_{r,k_r} z^b_{k_r,r}\} = w_b \ \forall b \in \mathcal{B}, && \text{(21e)}\\
&p_r, q_r, \bar{u}^j_{i,r}, \tilde{u}^j_r \geq 0,\ z^i_{j,r}, z^b_{j,r} \in \mathbb{Z} \ \forall i \in [\![n]\!], j \in [\![k]\!], r \in [\![N]\!], b \in \mathcal{B}. && \text{(21f)}
\end{aligned}
$$

A solution of (21) *yields the value of* z_{IP} *for every* $b \in \mathcal{D}$.

We can eliminate the round-down function by replacing (21b) and (21c) with:

$$
\begin{aligned}
z^i_{j,r} &\leq z^i_{1,r}\bar{u}^j_{i,r} + \cdots + z^i_{j-1,r}\bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r,\\
z^i_{j,r} &\geq z^1_{i,r}\bar{u}^j_{i,r} + \cdots + z^i_{j-1,r}\bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r - 1,\\
z^i_{j,r} &\in \mathbb{Z} \ \forall i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!].
\end{aligned} \tag{22}
$$

Next, in all of equations (21b), (21c), (21d) and (21e), the bilinear terms coming from dot product between $p, q, \bar{u}, \tilde{u}$ and z introduce non-linearity into the formulation. Nevertheless, we can also linearize this bilinear term relying on the boundedness of the variables z. For instance, given a lower bound and upper bound of $z^i_{j,r}$ for $i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!]$, we can model $q_{r,j} z^i_{j,r}$ as a piecewise linear function [42]. Finally, we can derive a MIP formulation for the minimum operator in (21e) following the approach in Anderson et al. [3, Section 5].

4.2 Computing CG Multipliers

In addition to allowing us to derive a MIP formulation for finding an IP value function, the representation of an IP value function via a CGNN can be viewed as a way of computing CG multipliers.

Corollary 4. *Let* $\mathrm{CGB}(\beta) : \mathbb{R}^m \to \mathbb{R}$ *be a function represented by a CG block with non-negative weights and round-down activation functions. If* $\mathrm{CGB}(a^i) > c_i \ \forall i \in [\![n]\!]$*, then* $\mathrm{CGB}(\beta)$ *is an upper bound of the IP value function* $z(\beta)$.

Hence, for a right-hand side b, finding the weights of a CG block that minimize $\mathrm{CGB}(b)$ gives us the optimal value of $IP(b)$. However, since each CG block represents one CG dual function, the weights of the CG block will be the CG multipliers that derive the convex hull of $IP(b)$. In general, we want to find the weight of a CG block that minimizes:

$$
\begin{aligned}
\min\ &\mathrm{CGB}(b)\\
\text{s.t}\ &\mathrm{CGB}(a^i) \geq c_i \ \forall i \in [\![n]\!].
\end{aligned} \tag{23}
$$

Even though solving (23) to optimality is difficult, obtaining any suboptimal solution where $\mathrm{CGB}(b) < z_{LP}(b)$ is meaningful because in this case, the weights of the CG block derive nontrivial CG inequalities.

4.3 Bounds on CG multipliers

In Theorem 3, we use the round-down operation as the activation function. When restricting the activation functions to ReLU, or other piecewise affine activation functions that only have a finite number of pieces, e.g., Leaky ReLU, binarized, or quantized activation functions [23, 45], we can only approximate the IP value function within a bounded domain. As the weight of a CGNN directly depends on the CG multipliers, bounds on the CG multipliers can derive bounds on the number of neurons in a CGNN with ReLU activation. Hence, in this subsection, we discuss a possible upper bound of the CG multipliers.

Certainly, when the right-hand side vector b varies, we may need different CG multipliers to derive the convex hull $S(b)$. Hence, in Definition 2, we use the superscript b to signal the dependence of a CG inequality on b. However, for the remaining of this subsection, we fix a right-hand side vector b and suppress the dependence on b for notations simplicity. For a vector $u \in \mathbb{R}^m_+$, we define $\{u\} \coloneqq [\{u_1\}, \dots, \{u_m\}]^T$ also be a vector in $\mathbb{R}^m_+$ of the fractional part of every element in u, that is $\{u\} = u - \lfloor u \rfloor$.

Lemma 7. *For any k non-negative vectors $u^1, \dots, u^k \in \mathbb{R}^m_+$, we have $\bar{P} \coloneqq \{x \in \mathbb{R}^n_+ | Ax \le b, \lfloor (u^i)^T A \rfloor x \le \lfloor (u^i)^T b \rfloor \ \forall i \in [\![k]\!]\}$ contains $\tilde{P} \coloneqq \{x \in \mathbb{R}^n_+ | Ax \le b, \lfloor \{u^i\}^T A \rfloor x \le \lfloor \{u^i\}^T b \rfloor \ \forall i \in [\![k]\!]\}$.*

Proof. For the base case, we show that for $u^1 \in \mathbb{R}^m_+$, $\bar{P}^1 \coloneqq \{x \in \mathbb{R}^n_+ | Ax \le b, \lfloor (u^1)^T A \rfloor x \le \lfloor (u^1)^T b \rfloor\}$ contains $\tilde{P}^1 \coloneqq \{x \in \mathbb{R}^n_+ | Ax \le b, \lfloor \{u^1\}^T A \rfloor x \le \lfloor \{u^1\}^T b \rfloor\}$.

For any $j \in [\![m]\!]$, we have

$$\begin{aligned} &\lfloor (u^1 - e_j)^T A \rfloor x \le \lfloor (u^1 - e_j)^T b \rfloor \\ \Leftrightarrow\ &\lfloor (u^1)^T A - A_j \rfloor x \le \lfloor (u^1)^T b - b_j \rfloor \\ \Leftrightarrow\ &\lfloor (u^1)^T A \rfloor x - A_j x \le \lfloor (u^1)^T b \rfloor - b_j, \end{aligned} \tag{24}$$

where e_j denotes the j^{th} unit vector. We obtain the last inequality because A_j - the j^{th} row of A - and b_i are integral. By taking sum of (24) and the i^{th} row of $Ax \le b$, we derive that $\lfloor u^T A \rfloor x \le \lfloor u^T b \rfloor$ is valid for $\tilde{P}$. By applying this procedure $\lfloor u^1_j \rfloor$ times for every $j \in [\![m]\!]$, we have $\tilde{P}^1 \subseteq \bar{P}^1$. By applying the same argument $k > 1$ times, we derive that $\tilde{P} \subseteq \bar{P}$.
In this section, we use S to denote $S(b)$ to suppress dependence on b when the context is clear. Since for any non-negative CG multiplier $u \in \mathbb{R}^m_+$, we always derive a valid inequality for S, thus $S \subseteq \tilde{P}$. Moreover, Lemma 7 states that we can replace the multipliers of every rank 1 CG inequality by their fractional parts and obtain a tighter relaxation. In what follows, we show that this still holds for higher-rank CG inequalities. Suppose that the convex hull S requires up to rank r CG inequalities ($r \in \mathbb{Z}_+$), we denote

$$\begin{aligned} u^1 = [u^1_1, \dots, u^1_{k_1}] &\text{ as multipliers corresponding to rank 1 CG inequalities,} \\ &\vdots \\ u^r = [u^r_1, \dots, u^r_{k_r}] &\text{ as multipliers corresponding to rank } r \text{ CG inequalities,} \end{aligned}$$

where each u^i is a matrix and u^i_j is a vector for every $j \in [\![k_i]\!]$, $i \in [\![r]\!]$, that defines linear constraints of S. Whenever we add new CG inequalities, we obtain a new LP with an updated constraint matrix and an updated right-hand side vector. Notationally, we let $A^0 \coloneqq A, b^0 \coloneqq b$, and

$$A^i = \begin{bmatrix} A^{i-1} \\ \lfloor (u^i)^T A^{i-1} \rfloor \end{bmatrix}, \text{ with } b^i = \begin{bmatrix} b^{i-1} \\ \lfloor (u^i)^T b^{i-1} \rfloor \end{bmatrix} \ \forall i \in [\![r]\!].$$

Similarly, we denote $S^0 \coloneqq \{x \in \mathbb{R}^n_+ | Ax \le b\}$ and $S^i \coloneqq \{x \in \mathbb{R}^n_+ | A_i x \le b_i,\}$ for $i \in [\![r]\!]$. For every $i \in [\![r]\!]$, S^i can be interpreted as the polyhedron where we add all rank i CG inequalities. Trivially, we have

$$S^0 \supseteq S^1 \supseteq \cdots \supseteq S^r = S.$$

We let $\tilde{u}^1_i = \{u^1_i\}$ for every $i \in [\![K_1]\!]$ and $\tilde{S}^1_b = \{x \in \mathbb{R}^n_+ | \tilde{A}^1 x \le \tilde{b}^1\}$, where $\tilde{A}_1$ and $\tilde{b}_1$ are constructed from A and b by introducing the CG inequalities corresponding to $\tilde{u}^i$ for every $i \in [\![k_1]\!]$. Based on Lemma 7, we have that $\tilde{S}_1 \subseteq S_1$. The main idea of the following theorem is that we want to construct a sequence $\tilde{S}^1 \supseteq \tilde{S}^2 \supseteq \cdots \supseteq \tilde{S}^r$ such that $\tilde{S}^i \subseteq S^i$ for every $i \in [\![r]\!]$ and thus $\tilde{S}^r = S$. □

Lemma 8. *For a positive integer $i \leq r$, suppose we have a polyhedron $\tilde{S}^i$ that satisfies $\tilde{S}^i \subseteq S^i$. Then we can construct a polyhedron $\tilde{S}^{i+1}$ from $\tilde{S}^i$ by adding CG inequalities with multipliers in* $[0, 1]$ *such that $\tilde{S}^{i+1} \subseteq S^{i+1}$.*

Proof. Since $\tilde{S}^i_b \subseteq S^i_b$, for every $k \in [\![k_i]\!]$, there exists v^i_k such that

$$(v^i_k)^T \tilde{A}^i = \lfloor (u^i_k)^T A^{i-1} \rfloor.$$

Hence, we can write A^i as a non-negative linear combination of rows in $\tilde{A}^i$, i.e., there exists V^i such that $V^i \tilde{A}^i = A^i$. Moreover, by construction, we have that:

$$A^{i+1} = \begin{bmatrix} A^i \\ \lfloor (u^{i+1}_1)^T A^i \rfloor \\ \vdots \\ \lfloor (u^{i+1}_{k_{i+1}})^T A^i \rfloor \end{bmatrix} = \begin{bmatrix} A_i \\ \lfloor (u^{i+1}_1)^T V^i \tilde{A}^i \rfloor \\ \vdots \\ \lfloor (u^{i+1}_{k_{i+1}})^T V^i \tilde{A}^i \rfloor \end{bmatrix}.$$

Let $\tilde{u}^{i+1}_l = \{(u^{i+1}_l)^T V^i\}$ for every $l \in [\![k_{i+1}]\!]$ and apply Lemma 7, we have:

$$\tilde{S}^{i+1} := \{x \in \mathbb{R}^n_+ | \tilde{A}^i x \leq \tilde{b}_i, \lfloor \tilde{u}^{i+1}_k \tilde{A}^i \rfloor x \leq \lfloor \tilde{u}^{i+1}_l \tilde{b}_i \rfloor \; \forall l \in [\![k_{i+1}]\!]\} \subseteq S^{i+1}_b.$$ □

By its construction, we have $\tilde{S}_1 \subseteq S_1$. We derive the following claim by applying Lemma 8. The following result can also be proven as a corollary from Theorem 7.2 of [15] and Lemma 7.

Theorem 4. *There exists a set of CG multipliers $\{u^i_l | l \in \|k_i\|, i \in [\![r]\!]\}$ corresponding to valid inequalities that defines $S(b)$, where r is the CG rank of $S(b)$, such that $\|u^i_j\|_\infty < 1$ for every $j \in [\![r_i]\!]$ and $i \in [\![r]\!]$.*

Proof. This is a direct consequence of Lemma 8. Since every valid inequality of $S(b)$ is a CG inequality, we derive that the CG inequality is obtained by multipliers of value between 0 and 1. □

5 Conclusion and Future Research

In this work, we have proved the existence of NNs that can approximate any IP value function within a desired L_1 tolerance. In addition to the NN Representation Theorems, our construction of IP value functions via CG multipliers can be used to derive a MIP formulation for the IP value functions over a possibly unbounded domain.

While we show that the set $\mathcal{B}$ in (21) contains a finite number of right-hand side vectors, obtaining every element of $\mathcal{B}$ is computationally expensive as there can be any exponential number of element in $\mathcal{B}$. We can replace the set $\mathcal{B}$ with any set of right-hand side vectors to look for a good approximation of the IP value function. The problem of finding good sub-optimal formulation for approximating the IP value function remains a subject for future research.

Acknowledgement. This work was supported by Office of Naval Research Grant N000142112262.

References

[1] R. M. Alfant, T. Ajayi, and A. J. Schaefer. Evaluating mixed-integer programming models over multiple right-hand sides. *Operations Research Letters*, 51(4):414–420, 2023.

[2] A. M. Alvarez, Q. Louveaux, and L. Wehenkel. A machine learning-based approximation of strong branching. *INFORMS Journal on Computing*, 29(1):185–195, 2017.

[3] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma. Strong mixed-integer programming formulations for trained neural networks. *Mathematical Programming*, 183(1):3–39, 2020.

[4] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv:1611.01491*, 2016.

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

[6] Y. Bengio, I. Goodfellow, and A. Courville. *Deep Learning*, volume 1. MIT press Cambridge, MA, USA, 2017.

[7] Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: A methodological tour d'Horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.

[8] D. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, 1996.

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

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

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

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

[13] Q. Cappart, D. Chételat, E. B. Khalil, A. Lodi, C. Morris, and P. Velickovic. Combinatorial optimization and reasoning with graph neural networks. *Journal of Machine Learning Research*, 24:130–1, 2023.

[14] C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming*, 83:451–464, 1998.

[15] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4): 305–337, 1973.

[16] W. Cook, A. M. H. Gerards, A. Schrijver, and E. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986.

[17] J. Dumouchelle, E. Julien, J. Kurtz, and E. B. Khalil. Neur2ro: Neural two-stage robust optimization. *arXiv:2310.04345*, 2023.

[18] A. Galassi, M. Lippi, and P. Torroni. Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308, 2020.

[19] R. E. Gomory. On the relation between integer and noninteger solutions to linear programs. *Proceedings of the National Academy of Sciences*, 53(2):260–265, 1965.

[20] B. Hanin. Universal function approximation by deep neural nets with bounded width and ReLU activations. *Mathematics*, 7(10):992, 2019.

[21] H. He, H. Daume III, and J. M. Eisner. Learning to search in branch and bound algorithms. *Advances in Neural Information Processing Systems*, 27, 2014.

[22] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[23] K. Huang, B. Ni, and X. Yang. Efficient quantization for neural networks with binary weights and low bitwidth activations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3854–3861, 2019.

[24] R. G. Jeroslow. Some basis theorems for integral monoids. *Mathematics of Operations Research*, 3(2): 145–154, 1978.

[25] H. Jia and S. Shen. Benders cut classification via support vector machines for solving two-stage stochastic programs. *INFORMS Journal on Optimization*, 3(3):278–297, 2021.

[26] N. Kong, A. J. Schaefer, and B. Hunsaker. Two-stage integer programs with stochastic right-hand sides: a superadditive dual approach. *Mathematical Programming*, 108(2):275–296, 2006.

[27] W. Kool, H. Van Hoof, and M. Welling. Attention, learn to solve routing problems! *arXiv:1803.08475*, 2018.

[28] E. Larsen, S. Lachapelle, Y. Bengio, E. Frejinger, S. Lacoste-Julien, and A. Lodi. Predicting tactical solutions to operational planning problems under imperfect information. *INFORMS Journal on Computing*, 34(1):227–242, 2022.

[29] E. Larsen, E. Frejinger, B. Gendron, and A. Lodi. Fast continuous and integer L-shaped heuristics through supervised learning. *INFORMS Journal on Computing*, 36(1):203–223, 2024.

[30] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[31] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33(12): 6999–7019, 2021.

[32] R. R. Meyer. On the existence of optimal solutions to integer and mixed-integer programming problems. *Mathematical Programming*, 7:223–235, 1974.

[33] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems*, 27, 2014.

[34] R. M. Patel, J. Dumouchelle, E. Khalil, and M. Bodur. Neur2sp: Neural two-stage stochastic programming. *Advances in Neural Information Processing Systems*, 35:23992–24005, 2022.

[35] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[36] A. Schrijver. On cutting planes. *Annals of Discrete Mathematics*, 9:291–296.

[37] Y. Tang, S. Agrawal, and Y. Faenza. Reinforcement learning for integer programming: Learning to cut. In *International Conference on Machine Learning*, pages 9367–9376. PMLR, 2020.

[38] K. M. Tarwani and S. Edem. Survey on recurrent neural network in natural language processing. *International Journal of Engineering Trends and Technology*, 48(6):301–304, 2017.

[39] O. Tavaslıoğlu, O. A. Prokopyev, and A. J. Schaefer. Solving stochastic and bilevel mixed-integer programs via a generalized value function. *Operations Research*, 67(6):1659–1677, 2019.

[40] A. C Trapp, O. A. Prokopyev, and A. J. Schaefer. On a level-set characterization of the value function of an integer program and its application to stochastic programming. *Operations Research*, 61(2):498–511, 2013.

[41] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5): 988–999, 1999.

[42] J. P. Vielma. Mixed-integer linear programming formulation techniques. *Siam Review*, 57(1):3–57, 2015.

[43] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[44] L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization*, volume 55. John Wiley & Sons, 1999.

[45] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu. Reluplex made more practical: Leaky ReLU. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–7. IEEE, 2020.