Representing Integer Program Value Function with Neural Networks

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

Abstract

We study the value function of an integer program (IP) by characterizing how its optimal value changes as the right-hand side varies. We show that the IP value function can be approximated to any desired degree of accuracy using machine learning (ML) techniques. Since an IP value function is a Chvátal-Gomory (CG) function, we first propose a neural network (NN) architecture as a universal approximator, which requires an explicit construction of the IP value function. We then derive a connection between CG cuts and the IP value function, which resulted in another NN architecture that does not require any information about the CG operations of the IP value function. Our novel NN architecture draws the relation between the weights of the NN and CG multipliers and inspired methods to derive a valid bound of the IP value function.

1 Introduction

Duality is a fundamental concept in optimization. While linear and convex programming duality are widely understood and critical for modern algorithms, less is known about integer programming (IP) duality. The seminal work of Blair and Jeroslow in the 1970s characterized the value function of a maximization IP, parameterized by its right-hand side, as a Gomory function, meaning that it can be constructed recursively from a set of linear functions through the composition of addition, nonnegative multiplication, minimization, and rounding down operations [7, 8, 9]. Despite its elegance, this result had a surprisingly limited impact on the computation of IP value function.

When discussing the approximation of functions, Machine Learning (ML) is widely recognized for its proficiency in this domain. As might be expected, ML has also received interest in the area of mathematical optimization [5]. Current research efforts in applying ML in discrete optimization primarily concentrate on developing policies that enable ML to make algorithmic decisions, such as node selection and branching decisions in branch-and-bound algorithms, or cut identification and classification [2, 16, 20, 27]. Alternatively, some research uses ML to directly estimate the solution of an IP from its input parameters [10, 21, 22, 30]. Furthermore, deep learning has frequently been utilized to approximate value functions for dynamic programming problems in the context of reinforcement learning, neuro-dynamic programming, or Markov Decision Processes [6, 29]. Typically, NNs - the primary models used in deep learning - are constructed through the recursive application of basic operations, such as affine transformations and piecewise linear activation functions (e.g., ReLU or

[*]Rice University. Email: tu.na@rice.edu
[†]Google. Email: joehuchette@google.com
[‡]Rice University. Email: ajs17@@rice.edu

max pooling). With sufficient layers and neurons in each layer, an NN can give a good approximation of any continuous function [4, 15, 17, 23]. While NNs have the universal approximation property, introducing bias or special architectures to an NN helps in certain applications, e.g., convolutional layers are typically used in image processing tasks [24], a recurrent NN or attention layer is one of the main component in natural language processing [13, 28]. In addition, certain constraints on the weights and architecture of an NN enforce that the NN always returns a convex function [3]. Based on these observations, a question arises: How can NN best be used to model IP value functions?

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

Operations	**CG Function**	**Neural Network**
Summation	Linear Combination	Affine Combination
Multivariate Non-linearity	Minimum over multiple inputs	Max-Pooling
Univariate Non-linearity	Round-down	ReLU, sign, etc

Table 1: Analogies between CG Functions and NN Construction.

In this work, inspired by the similarity between CG and NN operations, we introduce two representation theorems showing the possibility of NNs in approximating IP value functions:

T'ree Representation Theorem. Given an IP value function with at most k CG operations, defined on a bounded domain $\mathcal{D}$; Then, there exists a feed-forward NN of depth $O(k)$ and width $O(2^k)$ with ReLU activation function that approximates the IP value function within given $\epsilon > 0$ in the bounded domain $\mathcal{D}$.

One issue with the first representation theorem is that, while the coefficients of an IP value function are bounded, we usually do not know them in advance. The second representation theorem will address this issue:

Block Representation Theorem. Any rank r IP value function on a domain $\mathcal{D}$ (not necessarily bounded) can be represented by a feed-forward NN of depth $O(r)$ and bounded width (independent of the IP rank) 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 an NN that can approximate any IP value function within a given ϵ. Section 4 shows that the IP value function can be constructed via the CG cuts of a set of finite IPs. This result helps us derive the second NN architecture and method for obtaining the upper bound CG function of the true IP value function.

2 Structural Results

The NN construction is based on three CG operations that recursively construct the IP-value function. The motivations for the models in later sections on computing IP value functions are primarily based on the analogy between a CG function and an NN as pointed out in Table 1. Before stating the

main theorems, we describe results that support the main NN Representation Theorems. The main contribution in this section is the construction of the IP Value Function based on CG cuts.

2.1 Preliminaries

We demonstrate the ability of a NN to represent the IP value function of the form:

$$\begin{aligned} z(b) := \max\ & c^T x \\ \text{s.t } & Ax \leq b \\ & x \in \mathbb{Z}^n_+, \end{aligned} \tag{1}$$

where $c \in \mathbb{R}^n$ is a fixed objective, $A \in \mathbb{Z}^{m\times n}$ is a fixed integral constraint matrix with $(a^1, a^2, \ldots, a^n)$ denoting its columns. We denote $X(b) := \{x \in \mathbb{Z}^n_+ | Ax \leq b\}$ and b can vary and take any value in $\mathcal{D} = \{b \in \mathbb{Z}^m | X(b) \neq \emptyset\}$ - the set of feasible right-hand sides. If for some right-hand side vector b, the problem is unbounded above, we let $z(b) = +\infty$, while if the problem is infeasible, we let $z(b) = -\infty$. Throughout this paper, we use $IP(b)$ to denote the IP with the right-hand side vector b, and $LP(b)$ to denote its LP relaxation. We further assume that for every $b \in \mathcal{D}$, $IP(b)$ has a finite optimal solution. This is not a strict assumption as if $z(b) = +\infty$ for some right-hand side b, then there exists x^* such that $Ax^* \leq 0$ and $c^T x^* > 0$, which means $z(b) = +\infty$ for every $b \in \mathcal{D}$ [25].

Definition 1. ***(CG inequality).*** *[31] 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 a CG multiplier.*
2. *Construct a CG inequality $\sum_{j\in[\![n]\!]} \lfloor ua^j \rfloor x_j \leq \lfloor ub \rfloor$.*

When all the integer variables of $IP(b)$ can be bounded, the convex hull of $X(b)$, denoted as S_b, can be described by a finite number of CG inequalities [26]. The inequality $\sum_{j\in[\![n]\!]} \lfloor ua^j \rfloor x_j \leq \lfloor ub \rfloor$ can be added to the original set of constraints $Ax \leq b$ without changing the integral feasible region. Thus, we can apply the CG steps recursively again to obtain other CG inequalities.

Definition 2. ***(CG inequality rank).*** *[31] Since every valid inequality of S_b is a CG inequality [26], inductively, we say that a valid inequality $\pi^b x \leq \pi^b_0$ of S_b 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. The rank 0 CG inequalities are the valid inequalities of the feasible region of $LP(b)$, i.e., $\{x \in \mathbb{R}^n_+ | Ax \leq b\}$.*

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

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 the following. 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 $\epsilon \in (0,1)$, $h_\epsilon(x)$ 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 for 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

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

Thus,

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

□

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 derives the convex hull of the integral solution of $IP(b)$. 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 $\max\{c^T x | Ax \leq b, x \in \mathbb{Z}_+^n\}$ is equal to

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

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

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

Let $p^b, q_1^b, \ldots, q_r^b$ be an optimal solution of (4), and consider the following function:

Definition 3. *We say that $f_b(\cdot) : \mathbb{R}_+^m \to \mathbb{R}_+$ is a **CG dual function** with respect to the right-hand side vector b if for every $\beta \in \mathbb{Z}_+^m$,*

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

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 definition, 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 value function of an IP, 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 $f_b(\cdot) : \mathbb{R}^m_+ \to \mathbb{R}_+$, we have*

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

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

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 concerning the right-hand side b, we are obliged to incorporate all the necessary CG inequalities, thus solving the IP directly. 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. *Given 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,

$$tf_b(tb) = tf_b(b) = tz_{LP}(b) = z_{LP}(tb) \ \forall \ t \in \mathbb{R}_+.$$

□

Essentially, 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 $\max\{cx|Ax \leq b, x \in \mathbb{Z}^m_+\}$, i.e., f_b is a feasible solution of*

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

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

$$f_b(a^i) \geq c_i \ \forall i \in [\![n]\!].$$

Hence, the CG dual function f_b is a feasible solution to the superadditive dual. □

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 [31]:

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

where $D(b) \coloneqq \{d \in \mathbb{Z}_+^m | d \leq b\}$ and F is a vector with $|D(b)|$ coordinates. By the feasibility of f_b from Proposition 3, we have the following immediate connection between f_b and the LP (SDLP).

Corollary 1. *Given 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 $max\{cx | Ax \leq \beta, x \geq \mathbb{Z}_+^n\}$, we have*

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

Proof. By Proposition 3, we have $f_b(\beta) \geq z(\beta)$ for every $b, \beta \in \mathcal{D}$, thus $\min_{b \in \mathbb{Z}_+^m} f_b \geq z$. Moreover, by Proposition 1, when $b = \beta$, we must have $f_b(\beta) = z(\beta)$. Therefore, $z(\beta) = \min\{f_b(\beta) | b \in \mathbb{Z}_+^m\} \ \forall \beta \in \mathcal{D}$. □

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 of CG dual functions.

Example 1. *: In Figure 1, we consider an integer knapsack with 2 variables:*

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

Let $b = 1$ and solve the corresponding problem by adding a CG cut with the multiplier $u = 1/2$, we have 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_b (where $b = 1$) as

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

The plot of f_1 is given Figure 1. However, we can observe that the function f_1 itself is indeed the IP value function $z(b)$.

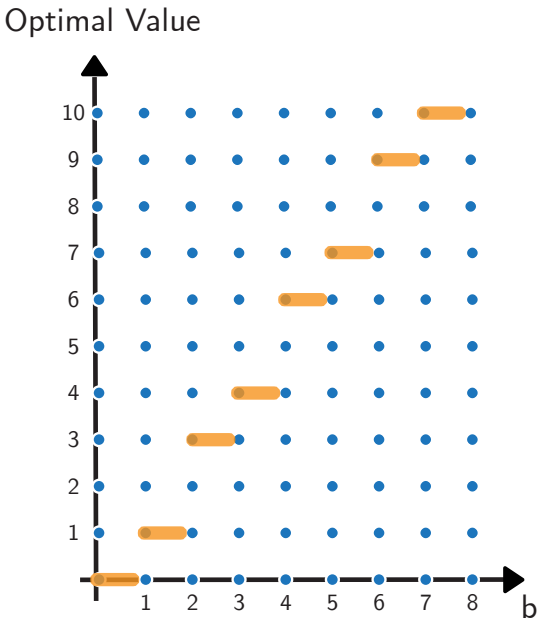

Figure 1: An illustration of a CG dual function in 1-dimensional space.

Example 2. *: We consider an IP with 2 constraints*

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

We construct two dual variable 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 2. When taking the minimum of these two functions, we obtain the IP value

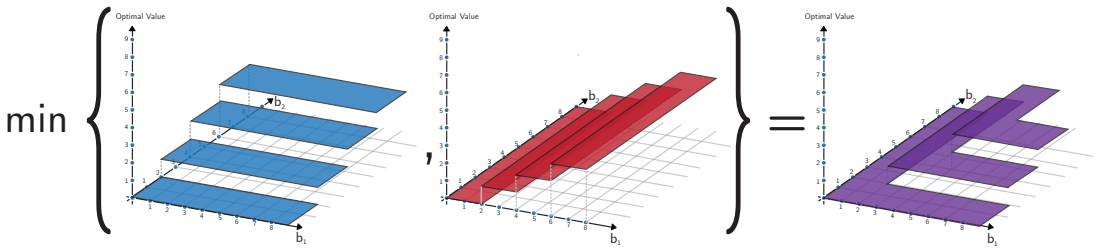

Figure 2: An illustration of the CG dual functions in 2-dimensional space.

function z in the rightmost.

As observed by these two examples, in constructing the IP value function, we might only need to take the minimum of a finite number of CG dual functions. In the next part of the section, we will prove that this conjecture is indeed true for every IP with a non-negative constraint matrix A.

2.4 Constructing The Integer Programming Value Function

In this subsection, we derive a "pattern" property of an IP value function. Generally speaking, we will show 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 Theorem 1 in [14]. 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] also came up with a way to compute IP value functions given known values of the function at some points $\hat{\beta} < \beta$ using the optimal solutions of $IP(\beta)$. However, in Proposition 3.2 [1], it is not guaranteed that the value function at every point in $\mathcal{D}$ can be computed using this property.

Consider the set $\mathcal{S} \coloneqq \{(c_1, a^1), \ldots, (c_n, a^n), (0, e_1), \ldots, (0, e_m), (-1, 0^m)\}$ and $\mathcal{C} \coloneqq \text{cone}(\mathcal{S})$ 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. By Lemma 3 in [19], for each face $F \in \mathcal{F}$, we have that the set of m extreme rays defining F is a subset of $\mathcal{S}$. 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 $\mathcal{F}'$.

Proposition 5. *Let $F \in \mathcal{F}'$ be a facet of $\mathcal{C}$, and let $\{(\gamma_1^F, v_1^F), \ldots (\gamma_m^F, v_m^F)\} \subset \mathcal{S}$ be the set of finite extreme rays defining F. Then, for every non-negative integers 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 matrix 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 $z(k_1 v_1^F + \cdots + k_m v_m^F) \geq k_1 \gamma_1^F + \cdots + k_m \gamma_m^F$, 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

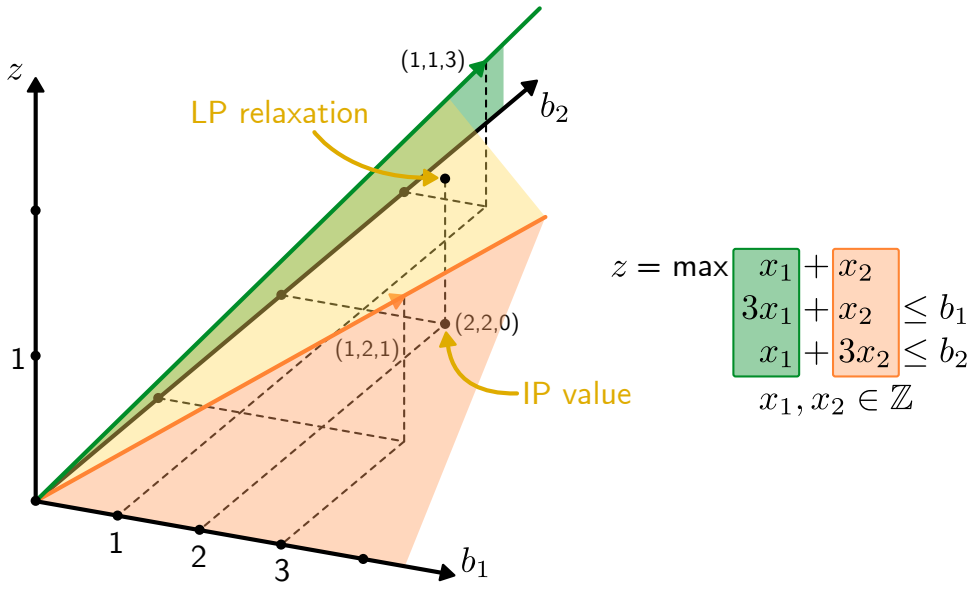

Figure 3: An illustration of cone $\mathcal{C}$. We only show here the 3 relevant facets of $\mathcal{C}$, while omitting the other 2 facets which contain the extreme rays $(-1,0,0)^T$. Intuitively, for a given right-hand side $b = (b_1, b_2)$, we move upward as far as possible until we reach a point 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.

$k_1\gamma_1^F + \cdots + k_m\gamma_m^F$ by setting the value of the variable corresponds to the column v_j^F equal to k_j. By contradiction, suppose that

$$z(k_1v_1^F + \cdots + k_mv_m^F) - (k_1\gamma_1^F + \cdots + k_m\gamma_m^F) > 0.$$

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

$$\epsilon \coloneqq z_{LP}(k_1v_1^F + \dots k_mv_m^F) - (k_1\gamma_1^F + \cdots + k_m\gamma_m^F) > 0.$$

Let x^* be the optimal primal solution of $zLP(k_1v_1^F + \cdots + k_mv_m^F)$, we have

$$\begin{cases} c^Tx^* - \epsilon = k_1\gamma_1^F + \cdots + k_m\gamma_m^F \\ Ax^* + I_ms^* = k_1v_1^F + \cdots + k_mv_m^F, \end{cases}$$

where $s^* = \sum_{i=1}^m k_iv_i^F - Ax^* \ge 0$. 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

$$z(k_1v_1^F + \dots k_mv_m^F) = k_1\gamma_1^F + \cdots + k_m\gamma_m^F.$$

□

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

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

Lemma 4. *Let $F \coloneqq cone((\gamma_1, v_1), \ldots, (\gamma_m, v_m)) \in \mathcal{F}'$ 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 $K_1, \ldots, K_m$ 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 $K_2, \ldots, K_m$ can be proved similarly. By Corollary 2 in [12], we have that $z_{LP}(b) - z(b) \leq M_{A,c}$, where the constant $M_{A,c}$ only depends on the constraint matrix A and the objective coefficient c. Therefore, we have that

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

The first inequality of Equation (11) follows from the fact that $z \leq z_{LP}$. The second inequality 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 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.,

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

We have that $z_{LP}(v_1) > f_0(v_1)$ because (γ_1, v_1) is an extreme ray of C. Let $\epsilon \coloneqq z_{LP}(v_1) - f_0(v_1)$ and $K_1 \coloneqq \lceil \frac{M_{A,c} + \gamma_1}{\epsilon} \rceil$. 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 $\bar{x}_{i^*} \geq 1$. By contradiction, suppose that $\bar{x}_{j^*} = 0$ in every optimal solution of $IP(\beta)$. We have

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

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

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

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 $\theta \geq 1$. 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 rays of F. Let $\mathcal{K} \coloneqq \max_{F \in \mathcal{F}', i \in [\![m]\!]} \|K_i^F v_i^F\|_1$. 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 $\beta = k_1^F v_1^F + \cdots + k_m^F v_m^F + \bar{b}$, where $\bar{b} \leq v_i^F$ for every $i \in [\![m]\!]$, there must exist $i \in [\![m]\!]$ such that $k_i^F > K_i^F$. Thus by Lemma 4, 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 a find face $F \in \mathcal{F}'$, and represent as $b = k_1^F v_1^F + \cdots + k_m^F v_m^F + \bar{b}$, we simply need to solve the relaxation $LP(\beta)$.

We can interpret Lemma 4 as a result of the "pattern" of the IP value function. i.e., when we know value of the function $z(\beta)$ at enough values of β, we can compute the value of z at different β based on the ones that we know. On the other hand, as we discussed earlier, for a fixed right-hand side vector b, there exists a set of CG multipliers that gives the convex hull of the feasible domain of $z(b)$. This observation and the "pattern" property from Lemma 4 raises a natural question of whether we can reuse the same CG multiplier for $z(\beta)$ for a different right-hand side $\beta \neq b$.

Lemma 5. *There exists a finite set $\mathcal{L}$ such that the function $l \coloneqq min\{f_b(\beta) | b \in \mathcal{L}\} \leq z_{LP}(\beta)$ is upper bounded by the LP value function z_{LP}, i.e.,*

$$l(\beta) \leq z_{LP}(\beta).$$

Proof. Since the value function of the LP relaxation z_{LP} 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 $l(\beta) \leq z_{LP}(\beta)$ for every $\beta \in \mathbb{R}^m$. □

Now, we use Lemma 4 and Lemma 5 to derive the main theorem of this section. The idea is to construct a function that has z_{LP} as an upper bound and the IP value function z as its lower bound. Then, we use the fact that the function z behaves in a pattern as described in Lemma 4, when our function agrees with z for enough number points, it must agree with z everywhere else.

Theorem 1. *There exists a finite set $\mathcal{B} \subset \mathbb{Z}_+^m$ such that*

$$z(\beta) = \min\{f_b(\beta) | b \in \mathcal{B}\} \ \forall \beta \in \mathbb{Z}_+^m.$$

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 4, and let $\mathcal{L}$ be the set of vectors that satisfies the condition stated in Lemma 5. Given the vectors b^F for every facet of F of C and $\mathcal{L}$, we choose $\mathcal{B} = \{b \in \mathbb{Z}_+^m | b \in \mathcal{L} \text{ or } \exists F \in \text{facet}(C) \text{ s.t } b \leq b^F\}$. We will show that the function $f^*(\beta) := \min\{f_b(\beta) | b \in \mathcal{B}\}$ equals to 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 $b \in \mathbb{Z}_+^m$ 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 4, 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. We have

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

Consider the univariate CG functions defined as follows $g^*(t) = f(\beta_1 + t\beta_2)$ for $t \in \mathbb{Z}$. Let c^* be the carrier of g^*. Since g^* is univariate, $g^*(0) = f^*(\beta_1)$, and g^* is non-decreasing, there must exist $\alpha \geq 0$ such that $c^*(t) = \alpha t + f^*(\beta_1)$. Since g^* is a CG function with a rational coefficient, we must have

$$0 \leq c^*(t) - g^*(t) \leq r^*,$$

where r^* is the CG rank of g^*, and $\gamma^*(t) := c^*(t) - g^*(t)$ is periodic. Let T denote the periodicity of γ^*, we have

$$\begin{aligned} & c^*(1+T) + T\gamma^*(1) + Tf^*(\beta_1) \geq (1+T)c^*(1) \\ \Leftrightarrow\ & c^*(1+T) - \gamma^*(1+T) + Tf^*(\beta_1) \geq (1+T)(c^*(1) - \gamma^*(1)) \\ \Leftrightarrow\ & f^*(\beta_1 + T\beta_2) + Tf^*(\beta_1) \geq (1+T)f^*(\beta_1 + \beta_2) \\ \Leftrightarrow\ & f^*(\beta_1 + (1+T)\beta_2) - f(\beta_1) \geq (1+T)(f^*(\beta_1 + \beta_2) - f^*(\beta_1)). \end{aligned}$$

We derive the first inequality based on the linearity of c^* and the fact that γ^* is always non-negative. For the second inequality, we subtract $(1+T)\gamma^*(1)$ from both sides and apply the periodic property. Intuitively, the final inequality tells us 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 $z(\beta) = f^*(\beta)\ \forall \beta \in \mathbb{Z}_+^m$. □

3 Tree Representation Theorem of the IP Value Function

This section describes how we can derive a NN structure that can approximate an IP value function. We respectively denote the three CG operators as:

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

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

$$\begin{aligned} F_h &\coloneqq F \cup \{h(f,g) | f, g \in F\}, \text{ if } h \in \{\Lambda^{\alpha,\beta}, \min\}, \\ F_h &\coloneqq F \cup \{\lfloor f \rfloor | f \in F\}, \text{ if } h = \lfloor \cdot \rfloor. \end{aligned}$$

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.

Lemma 6. *Let $\mathcal{G}_r$ be the class of Gomory functions of rank at most r. We have*

$$\mathcal{G}_r = \cup_{(h_1,\ldots,h_r) \subseteq \mathcal{H}^r} \mathcal{L}_{h_1,\ldots,h_r},$$

where $\mathcal{H}^r$ denotes the r-time Cartesian product for the set of CG operators $\mathcal{H}$ (when $r = 0$, $\mathcal{H}^0 = \emptyset$).

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 Gomory 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 $f' \in \mathcal{G}_k$. 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. $f = \min\{f', g'\}$ for some $f', g' \in \mathcal{G}_k$. By the induction hypothesis $f', g' \in \cup \mathcal{L}_{h_1,\ldots,h_r}$, thus $f \in \cup \mathcal{L}_{h_1,\ldots,h_r,\min}$.
3. $f = \alpha f' + \beta g'$ for some $\alpha, \beta \in \mathbb{R}_+$ and $f', g' \in \mathcal{G}_r$. By the induction hypothesis $f', g' \in \cup \mathcal{L}_{h_1,\ldots,h_r}$, thus $f \in \cup \mathcal{L}_{h_1,\ldots,h_r,\Lambda^{\alpha,\beta}}$.

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

Theorem 2. *Given a real number $\delta > 0$, a bounded input domain $\mathcal{B} \coloneqq \{b \in \mathcal{D} | \|b\|_1 \leq \mathcal{K}\}$, and a CG function $z(b)$ of rank r, there exists a NN f with $O(r)$ layers and $O(2^{r+1})$ neurons such that*

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

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 CG function that contains $r+1$ operations.

Case 1: Suppose $z = \alpha z_1 + \beta z_2$, where $\alpha, \beta \in \mathbb{R}_+$ 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

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

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

$$\int_{\mathcal{B}} |f - z| \leq \alpha \int_{\mathcal{B}} |f_1 - z_1| + \beta \int_{\mathcal{B}} |f_2 - z_2| \leq \delta.$$

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

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

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

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

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

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

Hence, we derive

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

□

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 CG operations of $z(b)$ allows us to assign each NN layer the corresponding operations.

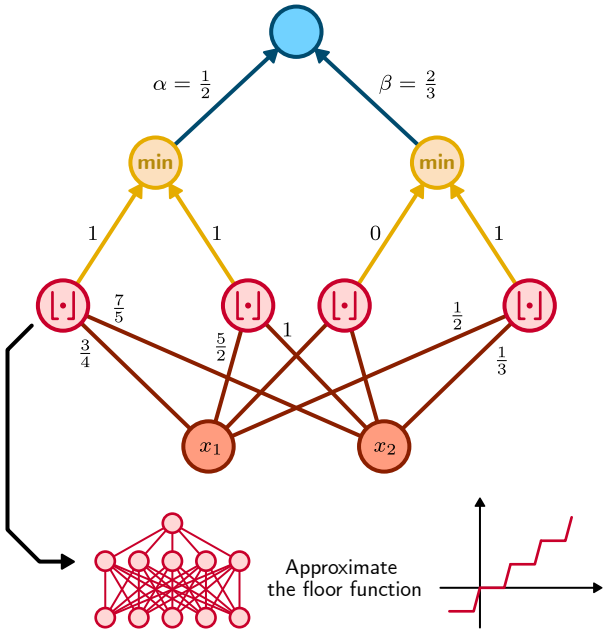

Figure 4: An example of the NN with exponential size width and linear size depth with respect to CG rank. Consider the function $f(x_1, x_2) = \frac{1}{2}\min\{\lfloor \frac{3}{4}x_1 + \frac{7}{5}x_2 \rfloor, \lfloor \frac{3}{2}x_1 + x_2 \rfloor\} + \frac{2}{3}\lfloor \frac{1}{2}x_1 + \frac{1}{3}x_2 \rfloor$. We can see that $f(x_1, x_2) \in \mathcal{L}_{\lfloor\cdot\rfloor,\min,\Lambda^{\alpha,\beta}}$. 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 $\Lambda^{\alpha,\beta}$ is linear, min is min, and $\lfloor\cdot\rfloor$ is the round-down function, which can be approximated using a smaller network.

We have shown the existence of an NN with a bounded size that approximates an IP value function. However, using this NN architecture cannot assure the monotone and superadditive property of an IP value function. In the next sections, we extend the structural results of Section 2 to derive another representation theorem, which allows us an NN training framework that guarantees an upper approximation of the IP value function.

4 Block Representation Theorem of the IP Value Function

Based on Theorem 1, we construct a NN structure that can capture the IP-value function $z(\beta)$. 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's architecture with $k + 1$ hidden neurons, where exactly one of them has no activation function, while the remaining k has an activation function that approximates the floor function. We use the term CG neuron to refer to a neuron that takes the right-hand side β and output of all previous neurons as input. In addition, we name an NN consisting only of CG neurons a CG-Block. When an NN is constructed by taking the minimum of multiple CG-Block, we call it a CG-Neural Network (or CGNN).

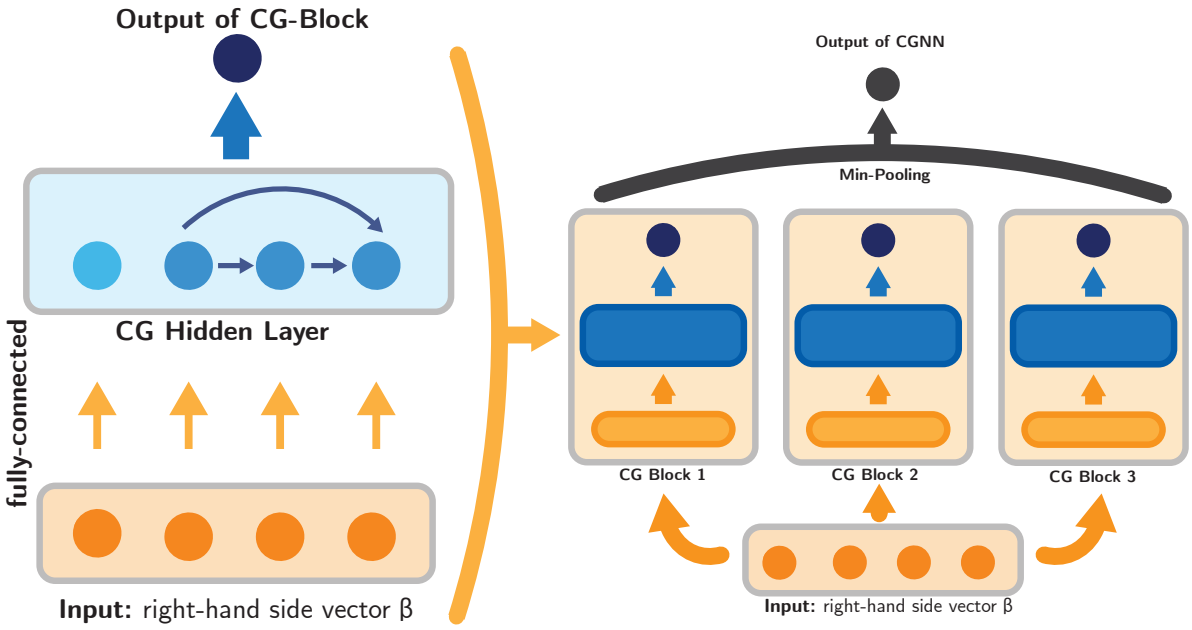

Figure 5: CGNN Architecture: (Left) An illustration of a CG-Block. (Right) An illustration of a CGNN containing multiple CG-Block and a Min-Pooling Layer.

Theorem 3. *There exists a finite set of N number $k_1, \ldots, k_N$ such that there exists a CGNN with N CG-blocks with k_i CG-neurons each that can represent 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 $z(\beta)$, i.e, $N = \|\mathcal{B}\|$. 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 k_i neurons to represent the CG dual function. □

4.1 Mixed-Integer Formulation for IP Value Function

In this subsection, we discuss an optimization formulation that guarantees a superadditive function which is an upper bound of 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 formulation for the superadditive dual feasibility of one block with k CG-neurons.

$$z_j^i = \lfloor z_i^1 \bar{u}_i^j + \cdots + z_i^{j-1} \bar{u}_{j-1}^j + a^i \cdot \tilde{u}^j \rfloor \ \forall i \in [\![n]\!], j \in [\![k]\!], \tag{15a}$$

$$a^i \cdot p + q_1 z_i^1 + \cdots + q_k z_i^k \geq c_i \ \forall i \in [\![n]\!], \tag{15b}$$

$$p, q, u \geq 0 \tag{15c}$$

In Model (15), we use the variables z for the post-activation values. Moreover, since for each CG-neurons, 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

can extend (15) for a superadditive dual feasible formulation to a CG net with N blocks.

$$z^i_{j,r} = \lfloor z^1_{i,r}\bar{u}^j_{i,r} + \cdots + z^{j-1}_i \bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r \rfloor \ \forall i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!] \tag{16a}$$

$$a^i \cdot p_r + q_{r,1} z^1_{i,r} + \cdots + q_{r,k_r} z^{k_r}_{i,r} \geq c_i \ \forall i \in [\![n]\!], r \in [\![N]\!], \tag{16b}$$

$$p, q, u \geq 0 \tag{16c}$$

In Model (16), we introduce variables p, q, u for each block (with the subscript $r \in [\![N]\!]$), where each block has k_r CG-neurons. According to Theorem 1, let $\mathcal{B}$ be the finite set that $z(\beta) = \min\{f_b(\beta) | b \in \mathcal{B}\}$, we have the following MIP for finding the IP value function.

$$\max \sum_{b \in \mathcal{B}} w_b \tag{17a}$$

$$z^i_{j,r} = \lfloor z^1_{i,r}\bar{u}^j_{i,r} + \cdots + z^{j-1}_i \bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r \rfloor \ \forall i \in [\![n]\!], j \in [\![k_r]\!], r \in [\![N]\!] \tag{17b}$$

$$z^b_{j,r} = \lfloor z^b_{i,r}\bar{u}^j_{i,r} + \cdots + z^{j-1}_i \bar{u}^j_{j-1,r} + a^i \cdot \tilde{u}^j_r \rfloor \ \forall b \in [\![\mathcal{B}]\!], j \in [\![k_r]\!], r \in [\![N]\!] \tag{17c}$$

$$a^i \cdot p_r + q_{r,1} z^1_{i,r} + \cdots + q_{r,k_r} z^{k_r}_{i,r} \geq c_i \ \forall i \in [\![n]\!], r \in [\![N]\!], \tag{17d}$$

$$a^i \cdot b + q_{r,1} z^1_{i,r} + \cdots + q_{r,k_r} z^{k_r}_{i,r} \geq w_b \ \forall b \in \mathcal{B}, r \in [\![N]\!], \tag{17e}$$

$$p, q, u \geq 0. \tag{17f}$$

Corollary 2. *A solution of (17) yields the value of z_{IP} for every $b \in \mathcal{D}$.*

4.2 Learning CG Multipliers

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

Corollary 3. *Let $CGB(\beta) : \mathbb{R}^m \to \mathbb{R}$ be a function represented by a CG-Block with non-negative weights and round-down activation functions. If $CGB(a^i) > c_i \forall i \in [\![n]\!]$, then $CBG(\beta)$ 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 $CGB(b)$ 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:

$$\begin{aligned} &\min \ CGB(b) \\ &\text{s.t } CGB(a^i) \geq c_i \ \forall i \in [\![n]\!]. \end{aligned} \tag{18}$$

Even though solving (18) to optimality is difficult, obtaining any suboptimal solution where $CGB(b) < z_{LP}(b)$ 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 [18, 32], we can

only approximate the IP value function within a bounded domain. Hence, in this subsection, we discuss a possible upper bound of the CG multipliers. 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. 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 $\bar{P} \coloneqq \{x \in \mathbb{R}^n_+ | Ax \leq b, \lfloor (u^i)^T A \rfloor x \leq \lfloor (u^i)^T b \rfloor \ \forall i \in [\![k]\!]\}$ contains $\tilde{P} \coloneqq \{x \in \mathbb{R}^n_+ | Ax \leq b, \lfloor \{u^i\}^T A \rfloor x \leq \lfloor \{u^i\}^T b \rfloor \ \forall i \in [\![k]\!]\}$.*

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

For any $j \in [\![m]\!]$ and e_j denotes the j^{th} unit vector, we have

$$\begin{aligned} & \lfloor (u^1 - e_j)^T A \rfloor x \leq \lfloor (u^1 - e_j)^T b \rfloor \\ \Leftrightarrow\ & \lfloor (u^1)^T A - A_j \rfloor x \leq \lfloor (u^1)^T b - b_j \rfloor \\ \Leftrightarrow\ & \lfloor (u^1)^T A \rfloor x - A_j x \leq \lfloor (u^1)^T b \rfloor - b_j. \end{aligned} \tag{19}$$

We obtain the last inequality because A_j - the j^{th} row of A and b_i are integral. By taking sum of (19) and the i^{th} row of $Ax \leq b$, we derive that $\lfloor u^T A \rfloor x \leq \lfloor u^T b \rfloor$ 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 for $k > 1$ 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 Chvátal-Gomory 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 $S^0 \coloneqq \{x \in \mathbb{R}^n_+ | Ax \leq b\}$ and $S^i \coloneqq \{x \in \mathbb{R}^n_+ | A_i x \leq b_i, \}$ 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 $\tilde{u}_i^1 = \{u_i^1\}$ for every $i \in [\![K_1]\!]$ and $\tilde{S}_b^1 = \{x \in \mathbb{R}_+^n | \tilde{A}^1 x \leq \tilde{b}^1\}$, 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 $\tilde{S}_b^i \subseteq S_b^i$, for every $k \in [\![k_i]\!]$, there exists v_k^i such that

$$(v_k^i)^T \tilde{A}^i = \lfloor (u_k^i)^T A^{i-1} \rfloor.$$

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:

$$A^{i+1} = \begin{bmatrix} A^i \\ \lfloor (u_1^{i+1})^T A^i \rfloor \\ \vdots \\ \lfloor (u_{k_{i+1}}^{i+1})^T A^i \rfloor \end{bmatrix} = \begin{bmatrix} A_i \\ \lfloor (u_1^{i+1})^T V^i \tilde{A}^i \rfloor \\ \vdots \\ \lfloor (u_{k_{i+1}}^{i+1})^T V^i \tilde{A}^i \rfloor \end{bmatrix}.$$

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

$$\tilde{S}^{i+1} \coloneqq \{x \in \mathbb{R}_+^n | \tilde{A}^i x \leq \tilde{b}_i, \lfloor \tilde{u}_k^{i+1} \tilde{A}^i \rfloor x \leq \lfloor \tilde{u}_l^{i+1} \tilde{b}_i \rfloor \ \forall l \in [\![k_{i+1}]\!]\} \subseteq S_b^{i+1}.$$

□

Since by 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 [11] and Lemma 7

Theorem 4. *There exists a set of CG multipliers $\{u_l^i | l \in \|k_i\|, i \in [\![r]\!]\}$ corresponding to valid inequalities that defines S_b, where r is the CG rank of S_b, such that $\|u_j^i\|_\infty < 1$ 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 result on constructing 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 (17) only 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 $\mathcal{B}$. On the other hand, 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 inquiry into identifying a good sub-optimal formulation for approximating the IP value function remains a subject for future research.

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] B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

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

[5] 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.

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

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

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

[9] 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.

[10] 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.

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

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

[13] 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.

[14] 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.

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

[16] 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.

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

[18] 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.

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

[20] 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.

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

[22] 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.

[23] 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.

[24] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

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

[26] Alexander Schrijver et al. On cutting planes. *Combinatorics*, 79:291–296, 1980.

[27] 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.

[28] 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.

[29] 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.

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

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

[32] 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.