

# Locally Ideal Formulations for Piecewise Linear Functions with Indicator Variables<sup>☆</sup>

Srikrishna Sridhar<sup>a</sup>, Jeff Linderoth<sup>b</sup>, James Luedtke<sup>b</sup>

<sup>a</sup>*School of Computer Sciences, University of Wisconsin-Madison, 1209 Dayton Street, Madison, WI 53706*

<sup>b</sup>*Department of Industrial and Systems Engineering, University of Wisconsin-Madison, 1513 University Avenue Madison, WI 53706*

---

## Abstract

In this paper, we consider mixed integer linear programming (MIP) formulations for piecewise linear functions (PLFs) that are evaluated when an indicator variable is turned on. We describe modifications to standard MIP formulations for PLFs with desirable theoretical properties and superior computational performance in this context.

---

## 1. Introduction

Optimization problems involving piecewise linear functions (PLFs) appear in a wide range of applications. PLFs are frequently used to approximate nonlinear functions and to model cost functions involving economies of scale and fixed charges. Problems involving non-convex PLFs are commonly formulated as mixed integer programming (MIP) problems [1, 2, 3, 4, 5].

Consider a univariate PLF  $f : [B_0, B_n] \rightarrow \mathbb{R}$  with its domain  $[B_0, B_n]$  divided into an increasing sequence of breakpoints  $\{B_0, B_1, \dots, B_n\}$ . For simplicity, we assume that  $f(\cdot)$  is continuous,  $B_0 = 0$  and  $f(0) = 0$ . Our results can be extended to the case when  $f(\cdot)$  is lower semi-continuous,  $B_0 \neq 0$ , and  $f(B_0) \neq 0$ . The function  $f(\cdot)$  can be written as

$$f(x) := m_i x + c_i, \quad x \in [B_{i-1}, B_i] \quad \forall i \in \{1, \dots, n\} \quad (1)$$

where  $m_i \in \mathbb{R}, c_i \in \mathbb{R}$  and  $B_0 < B_1 < \dots < B_n$ .

In this paper, we present MIP formulations for PLFs where setting a binary indicator variable to zero forces the argument of the function of  $f(\cdot)$  to zero which in turn forces the function to take a zero value. In other words,

$$z = 0 \Rightarrow x = 0, f(x) = 0. \quad (2)$$

The goal of this work is to present a theoretical and computational comparison of MIP formulations that enforce the logical conditions in (2). Specifically, we examine properties of different formulations of the three variable set

$$X := \bigcup_{i=1}^n \{(x, y, z) : x \in [B_{i-1}, B_i], y = m_i x + c_i, z = 1\} \cup \{(0, 0, 0)\}. \quad (3)$$

---

<sup>☆</sup>This work is supported through a contract from Argonne, a U.S. Department of Energy Office of Science laboratory. This work was supported by the Applied Mathematics activity, Advance Scientific Computing Research program within the DOE Office of Science.

In some applications, notably those where the PLF appears in a minimization objective, the relevant set to study has the variable  $y$  constrained to lie in the epigraph of a convex function. We denote  $X^{\geq}$  as the set where the equality relationship in (3) is replaced with  $y \geq m_i x + c_i$ .

Methods for modeling PLFs include specially ordered sets of type II (SOS2) [1], the incremental model, or delta method (Delta) [2], the multiple choice model (MCM) [6], the convex combination (CC) model [3], the disaggregated convex combination model (DCC) [7], and approaches that require only logarithmically many binary variables [8]. Table 1 lists several applications in the literature that have modeled PLFs using these well-known methods in conjunction with variable upper bound constraints of the form

$$x \leq B_n z \tag{4}$$

to enforce the logical on-off condition (2).

Table 1: Applications Using PLFs with Indicator Variables

Ref.	Application	Model
[9]	Gas network optimization	SOS2
[10]	Transmissions expansion planning	Delta
[11]	Oil field development	CC
[12]	Thermal unit commitment	Delta
[13]	Sales resource allocation	MCM

In this work, we propose a simple modeling artifice for PLFs that also enforces the logical condition (2), and we demonstrate its desirable theoretical and computational properties. We start by describing the idea using SOS2 to model a PLF as

$$x = \sum_{i=0}^n \lambda_i B_i, \quad y = \sum_{i=0}^n \lambda_i F_i, \quad 1 = \sum_{i=0}^n \lambda_i, \quad \boldsymbol{\lambda} := \left\{ \lambda_i \in \mathbb{R}_+ : \forall i \in \{0, \dots, n\} \right\} \text{ is SOS2.} \tag{5}$$

In this formulation, the function  $f(\cdot)$  and its argument  $x$  are expressed as convex combinations of breakpoints  $\mathcal{B} := \{B_0 \dots B_n\}$  and their corresponding function evaluations  $\{F_0 \dots F_n\}$  where  $F_i := f(B_i) = m_i B_i + c_i$ . The formulation introduces a non-negative set of variables  $\boldsymbol{\lambda} \in \mathbb{R}^{n+1}$  that satisfy the SOS2 property—at most two of the variables can be positive, and if two variables are positive then they must be consecutive in the ordered set. Most modern general purpose MIP solvers enforce the SOS2 condition algorithmically by branching [1].

Using variable upper bound constraints (4) to enforce the logical condition (2) has two problems. First, the use of “bigM” constraints may considerably weaken the LP relaxation of the MIP formulation. Second, the model introduces an additional constraint  $x \leq B_n z$ .

We propose the following simple strengthening that replaces  $x \leq B_n z$  and  $\sum_{i=0}^n \lambda_i = 1$  with

$$\sum_{i=0}^n \lambda_i = z. \tag{6}$$

Setting the binary variable  $z = 0$  in (6) forces  $\lambda_i = 0 \forall i \in \{0, \dots, n\}$ , which in turn forces the function to take a zero value. If the binary variable  $z = 1$ , then  $\sum_{i=0}^n \lambda_i = 1$ , which reduces to (5). We show in Section 2.1 that a formulation using (6) has the desirable property of being *locally ideal*, while one that uses  $x \leq B_n z$  does not.

In Section 2, we also show how to strengthen MIP formulations of  $X$  that use the incremental model, the multiple choice model, the convex combination model, the disaggregated convex combination model, and

logarithmic models to model the PLF. Therefore, this formulation strengthening technique could be directly applied to all of the applications listed in Table 1. In all cases, we show that our model retains the desirable theoretical property of the underlying PLF modeling method, either idealness or sharpness, but using a variable upper bound constraint  $x \leq B_n z$  destroys the property. Borghetti et al. [14] created a formulation of  $X$  that employed the strengthening techniques we describe. They used the convex combination method to model the PLFs which does not have the locally ideal property [5]. In the case that the PLFs are convex, we describe a connection between the formulation strengthening techniques we describe and the *perspective reformulation* [15].

We conclude with a computational study on a practical application to illustrate the benefits of the new formulations. In our experiments, we observed that our formulation computes optimal solutions on average 40 times faster.

## 2. Properties of MIP formulations

Padberg and Rijal [16] define a *locally ideal* MIP formulation as one where the vertices of its corresponding LP relaxation satisfy all required integrality conditions. Extending this definition, Croxton et al. [17] and Keha et al. [18] define a locally ideal SOS2 formulation as one whose LP relaxation has extreme points that all satisfying the SOS2 property. As shown by Vielma et al. [5], all commonly used MIP formulations of PLFs, except for the original convex combination (CC) model, are known to be locally ideal. In this section, we demonstrate the theoretical strength of proposed formulations for  $X$  that include the logical condition (2).

### 2.1. SOS2 Model

We consider the following two SOS2-based formulations for  $X$ :

$$S_1 := \left\{ (x, y, \boldsymbol{\lambda}, z) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+^{n+1} \times \{0, 1\} : x = \sum_{i=0}^n B_i \lambda_i, y = \sum_{i=0}^n F_i \lambda_i, 1 = \sum_{i=0}^n \lambda_i, x \leq B_n z, \boldsymbol{\lambda} \text{ is SOS2} \right\}$$

$$S_2 := \left\{ (x, y, \boldsymbol{\lambda}, z) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+^{n+1} \times \{0, 1\} : x = \sum_{i=0}^n B_i \lambda_i, y = \sum_{i=0}^n F_i \lambda_i, z = \sum_{i=0}^n \lambda_i, \boldsymbol{\lambda} \text{ is SOS2} \right\}$$

where  $S_1$  is a standard SOS2 model for PLFs that uses the constraint (4), while formulation  $S_2$  uses the constraint (6) to model the logical condition (2). One can easily show that both  $S_1$  and  $S_2$  are valid formulations of  $X$ . In other words, for either  $T = S_1$  or  $T = S_2$ ,

$$X = \left\{ (x, y, z) : \exists \boldsymbol{\lambda} \in [0, 1]^{n+1} \text{ s.t. } (x, y, z, \boldsymbol{\lambda}) \in T \right\}.$$

We use the standard definition of the *linear programming* (LP) relaxation of a model as the relaxation obtained by replacing integrality restrictions on variables with simple bound restrictions and by removing adjacency requirements for SOS2 variables. We now that prove that the formulation  $S_2$  is locally ideal while  $S_1$  is not.

**Theorem 1.** *Formulation  $S_2$  is locally ideal.*

*Proof.* The LP relaxation of  $S_2$  has  $n + 4$  variables, three equality constraints

$$x = \sum_{i=0}^n B_i \lambda_i, y = \sum_{i=0}^n F_i \lambda_i, z = \sum_{i=0}^n \lambda_i,$$

and  $n + 2$  inequality constraints,  $z \leq 1$  and  $\lambda_i \geq 0 \forall i = 0, 1, \dots, n$ . Extreme points of the LP relaxation of  $S_2$  have  $n + 4$  binding constraints, which forces at least  $n$  variables from  $\boldsymbol{\lambda} \in \mathbb{R}_+^{n+1}$  to be exactly equal to zero. Thus, the extreme points of the LP relaxation of  $S_2$  are

$$\{(x = B_i, y = F_i, \boldsymbol{\lambda} = B_i \vec{e}_i, z = 1) \forall i \in \{1, \dots, n\}\} \cup (x = 0, y = 0, \boldsymbol{\lambda} = \vec{0}, z = 0), \quad (7)$$

where  $\vec{e}_i$  are the  $n$  dimensional unit vectors. All points in (7) have  $z \in \{0, 1\}$  and satisfy the SOS2 properties for the  $\boldsymbol{\lambda}$  variables. Hence,  $S_2$  is locally ideal.  $\square$

A point  $(x, y, \boldsymbol{\lambda}, z)$  can only be an extreme point of the set

$$P_2^{\geq} := \left\{ (x, y, \boldsymbol{\lambda}, z) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+^{n+1} \times [0, 1] : x = \sum_{i=0}^n B_i \lambda_i, y \geq \sum_{i=0}^n F_i \lambda_i, z = \sum_{i=0}^n \lambda_i \right\}$$

if  $y = \sum_{i=0}^n F_i \lambda_i$ . Therefore, the proof of Theorem 1 also establishes that expressing logical condition (2) using (6) also results in a locally ideal formulation of  $X^{\geq}$ . Similar logic applies in our subsequent proofs of the local idealness of other formulations of  $X$  (Theorems 4 and 6). In each case, our proposed modeling of the logical condition (2) also yields a locally ideal formulation of  $X^{\geq}$ .

**Theorem 2.** *Formulation  $S_1$  is not locally ideal.*

*Proof.* Consider an instance with  $n = 3$ ,  $\mathbf{B} = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ , and  $\mathbf{F} = \{0, 4, 2, 3\}$ . The point  $\{x = \frac{1}{3}, y = 4, \boldsymbol{\lambda} = (0, 1, 0, 0), z = \frac{1}{3}\}$  is feasible to the LP relaxation of  $S_1$  but not feasible for the LP relaxation of  $S_2$ . Since the LP relaxation of  $S_2$  is a subset of the LP relaxation of  $S_1$ ,  $S_1$  cannot be locally ideal.  $\square$

An interesting consequence of Theorem 1 is that when the PLF is convex, the application of the reformulation technique we suggest to the set  $X^{\geq}$  is equivalent to the *perspective reformulation* [15], a preprocessing technique for (convex) mixed integer nonlinear programs that have the logical indicator structure (2). If  $f(\cdot)$  is convex, then  $m_1 > m_2 > \dots > m_n$ , and the perspective reformulation of  $X^{\geq}$  is

$$P = \{(x, y, z) \in \mathbb{R}^2 \times [0, 1] : y \geq m_i x + c_i z \forall i \in \{1, \dots, n\}, 0 \leq x \leq B_n z\},$$

where  $m_i := (F_i - F_{i-1}) / (B_i - B_{i-1})$  and  $c_i := (F_{i-1} - B_{i-1}(F_i - F_{i-1})) / (B_i - B_{i-1})$ . Günlük and Linderoth [19] show that if  $f(\cdot)$  is convex, then  $P = \text{conv}(X^{\geq})$ . The formulation  $S_2$  is locally ideal, so  $P_2^{\geq}$  must also be a formulation that is similarly strong.

**Corollary 3.**  $\text{Proj}_{xyz}(P_2^{\geq}) = P = \text{conv}(X^{\geq})$

## 2.2. Incremental Model

The incremental model introduces a set of non-negative variables  $\boldsymbol{\delta} := \{\delta_1, \dots, \delta_n\}$  to model the portion of each interval “filled” by the variable  $x$ . The interval  $i + 1$  can be filled ( $\delta_{i+1} > 0$ ) only if the interval  $i$  is already filled ( $\delta_i = 1$ ). Unlike the SOS2 model, the incremental model specifically requires the introduction of binary variables  $\mathbf{b} \in \{0, 1\}^{n-1}$  to enforce the necessary ordering conditions. To model the on-off logical condition (2), the incremental model can be augmented with a variable upper bound constraint  $x \leq B_n z$ , resulting in a formulation

$$\Delta_1 := \left\{ (x, y, \boldsymbol{\delta}, z, \mathbf{b}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \{0, 1\} \times \{0, 1\}^{n-1} : x = \sum_{i=1}^n [B_i - B_{i-1}] \delta_i, y = \sum_{i=1}^n [F_i - F_{i-1}] \delta_i, x \leq B_n z, \right. \\ \left. \delta_1 \leq 1, 0 \leq \delta_n, \delta_{i+1} \leq b_i \leq \delta_i \forall i \in \{1, \dots, n-1\} \right\}.$$

Alternatively, the on-off condition can be enforced by replacing the constraint  $\delta_1 \leq 1$  with  $\delta_1 \leq z$ , yielding the formulation

$$\Delta_2 := \left\{ (x, y, \boldsymbol{\delta}, z, \mathbf{b}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \{0, 1\} \times \{0, 1\}^{n-1} : x = \sum_{i=1}^n [B_i - B_{i-1}] \delta_i, y = \sum_{i=1}^n [F_i - F_{i-1}] \delta_i, \right. \\ \left. \delta_1 \leq z, 0 \leq \delta_n, \delta_{i+1} \leq b_i \leq \delta_i \forall i \in \{1, \dots, n-1\} \right\}.$$

Incremental models that use  $\delta_1 \leq z$  are locally ideal, while those that use  $x \leq B_n z$  are not.

**Theorem 4.** *Formulation  $\Delta_2$  is locally ideal.*

*Proof.* The matrix for the constraint system in  $\Delta_2$ , ignoring the constraints defining  $x$  and  $y$ , is

$$\begin{array}{rcl} -\delta_1 & +z & \geq 0, \\ \delta_i - b_i & & \geq 0 \quad \forall i \in \{1, \dots, n-1\}, \\ -\delta_{i+1} + b_i & & \geq 0 \quad \forall i \in \{1, \dots, n-1\}, \\ \delta_n & & \geq 0, \end{array}$$

which is a network matrix, and hence is totally unimodular. Thus all extreme points of the LP relaxation of  $\Delta_2$  naturally satisfy the requisite integrality properties.  $\square$

**Theorem 5.** *Formulation  $\Delta_1$  is not locally ideal.*

*Proof.* Consider an instance with  $n = 3$ ,  $\mathbf{B} = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$  and  $f(\mathbf{B}) = \{0, 4, 2, 3\}$ . The fractional point  $\{x = \frac{1}{3}, y = 4, \boldsymbol{\delta} = (1, 0, 0), z = \frac{1}{3}, \mathbf{b} = (0, 0)\}$  is feasible to the LP relaxation of  $\Delta_1$  but not feasible for the LP relaxation of  $\Delta_2$ .  $\square$

### 2.3. Multiple choice model

In the multiple choice model, a non-negative set of variables  $\mathbf{w} := \{w_1, \dots, w_n\}$  and an additional set of binary variable  $\mathbf{b} := \{b_1, \dots, b_n\}$  are introduced, with the logical implication that  $w_i = x$  if  $x$  is in the  $i$ th interval, and  $w_i = 0$  otherwise. Using a variable upper bound constraint to enforce the logical condition (2) with the multiple choice model gives the following formulation of  $X$ :

$$M_1 := \left\{ (x, y, \mathbf{w}, z, \mathbf{b}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \{0, 1\} \times \{0, 1\}^n : \sum_{i=1}^n w_i = x, y = \sum_{i=1}^n (m_i w_i + c_i b_i), x \leq B_n z, \right. \\ \left. \sum_{i=1}^n b_i = 1, B_{i-1} b_i \leq w_i \leq B_i b_i \forall i \in \{1, \dots, n\} \right\}.$$

Instead, the on-off condition can be formulated by replacing the constraints  $\sum_{i=1}^n b_i = 1$  with  $\sum_{i=1}^n b_i = z$ , yielding a formulation

$$M_2 := \left\{ (x, y, \mathbf{w}, z, \mathbf{b}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^n \times \{0, 1\} \times \{0, 1\}^n : \sum_{i=1}^n w_i = x, y = \sum_{i=1}^n (m_i w_i + c_i b_i), \right. \\ \left. \sum_{i=1}^n b_i = z, B_{i-1} b_i \leq w_i \leq B_i b_i \forall i \in \{1, \dots, n\} \right\}.$$

**Theorem 6.** *Formulation  $M_2$  is locally ideal.*

*Proof.* Following Balas [20], we write an extended formulation for the convex hull of the union of the  $n+1$  polytopes  $X_0 = \{(0, 0, 0)\}$ ,  $X_i = \{(x, y, z) : B_{i-1} \leq x \leq B_i, y = m_i x + c_i, z = 1\} \forall i \in \{1, \dots, n\}$  as those

$(x, y, z)$  for which there exist vectors  $\mathbf{w} = [w_0, \dots, w_n]$ ,  $\mathbf{v} = [v_0, \dots, v_n]$ ,  $\mathbf{u} = [u_0, \dots, u_n]$ ,  $\mathbf{b} = [b_0, \dots, b_n]$  such that the following inequality system is satisfied:

$$x = \sum_{i=0}^n w_i, \quad y = \sum_{i=0}^n v_i, \quad z = \sum_{i=0}^n u_i, \quad 1 = \sum_{i=0}^n b_i, \quad w_0 = 0, \quad v_0 = 0, \quad u_0 = 0, \quad b_i \geq 0 \quad \forall i \in \{0, \dots, n\}$$

$$B_{i-1}b_i \leq w_i \leq B_i b_i \quad \forall i \in \{1, \dots, n\},$$

$$v_i = m_i w_i + c_i b_i \quad \forall i \in \{1, \dots, n\},$$

$$u_i = b_i \quad \forall i \in \{1, \dots, n\}.$$

We can eliminate  $b_0$ ,  $\mathbf{u}$ , and  $\mathbf{v}$  from this system to obtain

$$x = \sum_{i=1}^n w_i, \quad y = \sum_{i=1}^n (m_i w_i + c_i b_i), \quad z = \sum_{i=1}^n b_i, \quad z \leq 1, \quad b_i \geq 0, \quad B_{i-1}b_i \leq w_i \leq B_i b_i \quad \forall i \in \{1, \dots, n\},$$

which is equivalent to the LP relaxation of  $M_2$ . □

**Theorem 7.** *Formulation  $M_1$  is not locally ideal.*

*Proof.* Consider an instance with  $n = 3$ ,  $\mathbf{B} = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ , and  $f(\mathbf{B}) = \{0, 4, 2, 3\}$ . The point  $\{x = \frac{1}{3}, y = 4, w = (0, \frac{1}{3}, 0), z = \frac{1}{3}, b = (0, 1, 0)\}$  is feasible to the linear programming relaxation of  $M_1$ , but not feasible for  $M_2$ . □

#### 2.4. Convex Combination Model

Another popular formulation for PLFs is the convex combination model, also known as the lambda method. The convex combination model uses continuous variables  $\boldsymbol{\lambda} \in \mathbb{R}^{n+1}$  and binary variables  $\mathbf{b} \in \{0, 1\}^n$ . The continuous variables are used to express  $x$  and  $y$  in terms of the breakpoints  $\mathbf{B}$  and function values  $\mathbf{F}$ . The binary variables are used to enforce the adjacency condition that  $b_i = 1 \Rightarrow \lambda_j = 0, \forall j \notin \{i-1, i\}$ . Using a variable upper bound to model the logical on-off condition (2) in combination with the most commonly used convex combination model gives the following formulation of  $X$ :

$$C_1 := \left\{ (x, y, \boldsymbol{\lambda}, z, \mathbf{b}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+^{n+1} \times \{0, 1\} \times \{0, 1\}^n : x = \sum_{i=0}^n \lambda_i B_i, \quad y = \sum_{i=0}^n \lambda_i F_i, \quad x \leq B_n z, \quad \sum_{i=0}^n \lambda_i = 1, \right.$$

$$\left. \sum_{i=1}^n b_i = 1, \quad \lambda_0 \leq b_1, \quad \lambda_n \leq b_n, \quad \lambda_i \leq b_i + b_{i+1} \quad \forall i \in \{1 \dots n-1\} \right\}.$$

Instead, the on-off condition can be directly imposed by replacing  $\sum_{i=1}^n b_i = 1$  and  $\sum_{i=0}^n \lambda_i = 1$  with the constraints  $\sum_{i=1}^n b_i = \sum_{i=0}^n \lambda_i = z$ . This gives the following formulation of  $X$ :

$$C_2 := \left\{ (x, y, \boldsymbol{\lambda}, z, \mathbf{b}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}_+^{n+1} \times \{0, 1\} \times \{0, 1\}^n : x = \sum_{i=0}^n \lambda_i B_i, \quad y = \sum_{i=0}^n \lambda_i F_i, \quad \sum_{i=0}^n \lambda_i = z, \quad \sum_{i=1}^n b_i = z, \right.$$

$$\left. \lambda_0 \leq b_1, \quad \lambda_n \leq b_n, \quad \lambda_i \leq b_i + b_{i+1} \quad \forall i \in \{1 \dots n-1\} \right\}.$$

It has been shown by Padberg [21] and Lee and Wilson [22] that the convex combination model that uses the constraints

$$\lambda_0 \leq b_1, \quad \lambda_n \leq b_n, \quad \lambda_i \leq b_i + b_{i+1} \quad \forall i \in \{1 \dots n-1\} \tag{8}$$

to model adjacency is *not* locally ideal. Padberg [21] gives the following improved formulation of the adjacency conditions:

$$\sum_{i=j}^n \lambda_i \leq \sum_{i=j}^n b_i, \quad \sum_{i=0}^{j-1} \lambda_i \leq \sum_{i=1}^j b_i \quad \forall j = 1, \dots, n,$$

which does result in a locally ideal formulation of PLFs. However, in most presentations of the convex combination model in the literature [3, 14, 11, 23] the non-ideal formulation (8) is used.

The convex combination model with constraints (8) does not result in a formulation that is locally ideal, however it does satisfy *sharpness*, a slightly weaker desirable property. An extended MIP formulation of a convex set is *sharp* if the extreme points of the projection of the LP relaxation of the formulation to the original space of variables ( $x, y, z$  in this case) satisfy integrality [6]. Vielma et al. [5] showed that the convex combination model that uses adjacency constraint (8) is sharp. We now show that the formulation  $C_2$  is sharp while  $C_1$  is not sharp.

**Theorem 8.** *Formulation  $C_2$  is sharp.*

*Proof.* Suppose that  $t = (x, y, \boldsymbol{\lambda}, z, \mathbf{b})$  is an extreme point of the linear programming relaxation of  $C_2$  with  $0 < z < 1$ . For  $\epsilon > 0$  define the points  $t^+ = (x^+, y^+, \boldsymbol{\lambda}^+, z^+, \mathbf{b}^+)$  and  $t^- = (x^-, y^-, \boldsymbol{\lambda}^-, z^-, \mathbf{b}^-)$  as

$$b_i^+ = (1 + \epsilon)b_i, \lambda_i^+ = (1 + \epsilon)\lambda_i, \quad \forall i \in \{1, \dots, n\}, z^+ = (1 + \epsilon)z, \quad x^+ = \sum_{i=0}^n \lambda_i^+ B_i, \quad y^+ = \sum_{i=0}^n \lambda_i^+ F_i$$

$$b_i^- = (1 - \epsilon)b_i, \lambda_i^- = (1 - \epsilon)\lambda_i, \quad \forall i \in \{1, \dots, n\}, z^- = (1 - \epsilon)z, \quad x^- = \sum_{i=0}^n \lambda_i^- B_i, \quad y^- = \sum_{i=0}^n \lambda_i^- F_i.$$

For some  $\epsilon > 0$ , the points  $t^+, t^-$  are both feasible for the linear programming relaxation of  $C_2$ , and  $t = 0.5(t^+ + t^-)$ , so  $t$  must not have been an extreme point.  $\square$

**Theorem 9.** *Formulation  $C_1$  is not sharp.*

*Proof.* Consider an instance with  $n = 3$ ,  $\mathbf{B} = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$ , and  $f(\mathbf{B}) = \{0, 4, 2, 3\}$ . One can verify that one of extreme points of the projection of the linear programming relaxation of  $C_1$  is  $\{x = \frac{1}{3}, y = 4, z = \frac{1}{3}\}$ , which does not satisfy the required integrality constraint on  $z$ .  $\square$

## 2.5. Other formulations

The disaggregated convex combination model for PLFs uses two sets of non-negative variables  $\boldsymbol{\lambda} := \{\lambda_i \mid i \in \{1, \dots, n\}\}$  and  $\boldsymbol{\mu} := \{\mu_i \mid i \in \{1, \dots, n\}\}$  and a set of binary variables  $\mathbf{b} := \{b_i \mid i \in \{1, \dots, n\}\}$ . The disaggregated convex combination model for a PLF is

$$y = \sum_{i=1}^n (\lambda_i F_i + \mu_i F_{i-1}), \quad x = \sum_{i=1}^n (\lambda_i B_i + \mu_i B_{i-1}) \quad \sum_{i=1}^n b_i = 1, \quad b_i = \lambda_i + \mu_i \quad \forall i \in \{1, \dots, n\}. \quad (9)$$

This formulation can be extended to model  $X$  by replacing the constraints  $\sum_{i=1}^n b_i = 1$  with  $\sum_{i=1}^n b_i = z$ . Disaggregated convex combination models that use these constraints are a locally ideal formulation of  $X$ .

Vielma and Nemhauser [8] modify the disaggregated convex combination model to use a logarithmic number of binary variables. Using notation defined in Vielma and Nemhauser [8], replacing  $\sum_{i=1}^n \lambda_i = 1$  with  $\sum_{i=1}^n \lambda_i = z$  is a valid locally ideal reformulation of model  $X$ . For the sake of brevity, we have omitted detailed discussions and proofs concerning disaggregated convex combination models.

### 3. Computational Results

In this section, we illustrate with numerical experiments the impact of using a locally ideal formulation ( $S_2$ ) instead of a weaker model ( $S_1$ ) that is not locally ideal.

#### 3.1. Practical Application

To make the numerical comparison, we consider an advertising budget allocation problem introduced by Zoltners and Sinha [24]. In this problem, a company is required to allocate an advertising budget  $D$  among a set  $\mathcal{K}$  of advertising strategies for a set of  $\mathcal{P}$  products. Let  $x_{jk}$  denote the amount of the advertising resource allocated to strategy  $k \in \mathcal{K}$  for product  $j \in \mathcal{J}$ . The company incurs a fixed cost  $G_j$  for entering the market with product  $j \in \mathcal{J}$  as well as a variable cost  $c_{jk}$  for each unit of the resource allocated to strategy  $k \in \mathcal{K}$  of product  $j \in \mathcal{J}$ . The return on investment is evaluated by piecewise-linear functions  $y_{jk} = f_{jk}(x_{jk})$  which have the typical form shown in Figure 2.

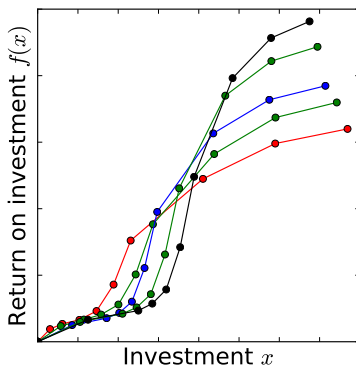


Figure 1: Sample curves modeling return on investment for five different product/strategy pairs.

A MIP formulation for this problem is

$$\begin{aligned}
 & \max \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{K}} y_{jk} \\
 & \sum_{i \in \mathcal{J}} \sum_{j \in \mathcal{K}} c_{jk} x_{jk} + \sum_{i \in \mathcal{J}} G_j z_j \leq D \\
 & (x_{jk}, y_{jk}, z_j) \in X_{jk} \quad \forall j \in \mathcal{J}, k \in \mathcal{K},
 \end{aligned} \tag{AP}$$

where  $X_{jk}$  is meant to denote that each of the triplets of variables  $(x_{jk}, y_{jk}, z_j)$  must lie in a set  $X$  (defined in (3)) specific to the product/strategy pair. We denote by  $AP(S_1)$  the MIP formulation of  $AP$  that uses  $S_1$  to model (3) and  $AP(S_2)$  as the MIP formulation of  $AP$  that uses the stronger formulation  $S_2$ .

#### 3.2. Numerical Results

We report tests conducted on 120 simulated instances of  $AP(X)$ . We created 20 random instances for each of the six problem sizes  $(|\mathcal{J}|, |\mathcal{K}|, n) \in \{(50, 50, 10), (50, 100, 10), (100, 100, 10), (50, 50, 20), (50, 100, 20), (100, 100, 20)\}$ . All instances were solved to 0.1% optimality using Gurobi 4.5.1 with default options on 2.66GHz Intel Core2 Quad CPU Q9400 processor with 8GB RAM. For all instances, we compare the quality of the LP relaxation as the percentage gap between the root LP relaxation value of the MIP formulations  $AP(S_2)$  and  $AP(S_1)$  relative to the optimal solution for each



Table 2: Summary of performance of formulations  $AP(S_2)$  and  $AP(S_1)$  on 120 simulated instances. Arithmetic mean, standard deviation, and geometric mean are shown.

Metric	Model	A.M	St. Dev	G.M
LP gap (%)	$AP(S_2)$	0.05	0.05	0.03
	$AP(S_1)$	19.7	1.5	19.6
Time (s)	$AP(S_2)$	16.8	12.0	12.3
	$AP(S_1)$	703.0	853.1	255.6
Nodes	$AP(S_2)$	26.3	33.1	9.2
	$AP(S_1)$	402.9	312.4	314.5

instance. We also measure the CPU time taken (using a single thread) and number of nodes in the search tree. Table 2 shows the summary statistics of our experiment.

The results convincingly demonstrate the advantage of using the locally ideal formulation  $AP(S_2)$ . The average root gap for  $AP(S_2)$  was 0.05%, while for  $AP(S_1)$  the average root gap was 19.6%. In fact, the best root gap for any instance of  $AP(S_1)$  was 17.1%. In terms of MIP solve times,  $AP(S_1)$  was solved on average in 703 seconds, while  $AP(S_2)$  was solved 41.8 times faster on average. In the worst case, Gurobi explored 1117 times more nodes on an instance modeled with  $AP(S_1)$  than with  $AP(S_2)$ . Clearly, one should use the locally ideal model  $AP(S_2)$ .

#### 4. Concluding remarks

In this paper, we present a theoretical and computational comparison of MIP models for PLFs where a binary indicator variable determines if the function is required to be evaluated. We propose strong formulations for this general class of MIP models by extending standard textbook PLF models including the incremental method, SOS2-based models, the multiple choice model, the convex combination model, and others. We showed in all cases that our formulations are either locally ideal or sharp, while a standard formulation that uses a variable upper bound constraint is not. Our numerical experiments demonstrate that our proposed formulations have significant computational advantages.

#### References

- [1] E. W. L. Beale, J. A. Tomlin, Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables, in: J. Lawrence (Ed.), Proceedings of the 5th International Conference on Operations Research, 1970, pp. 447–454.
- [2] H. M. Markowitz, A. S. Manne, On the Solution of Discrete Programming Problems, *Econometrica* 25 (1957) 84–110.
- [3] G. Dantzig, On the significance of solving linear programming problems with some integer variables, *Econometrica*, *Journal of the Econometric Society* 28 (1960) 30–44.
- [4] A. Balakrishnan, S. C. Graves, A composite algorithm for a concave-cost network flow problem, *Networks* 19 (1989) 175–202.
- [5] J. P. Vielma, S. Ahmed, G. Nemhauser, Mixed-Integer Models for Nonseparable Piecewise-Linear Optimization: Unifying Framework and Extensions, *Operations Research* 58 (2009) 303–315.

- [6] R. G. Jeroslow, J. K. Lowe, Modelling with integer variables, in: *Mathematical Programming at Oberwolfach II*, volume 22 of *Mathematical Programming Studies*, Springer Berlin Heidelberg, 1984, pp. 167–184. 10.1007/BFb0121015.
- [7] R. R. Meyer, Integer and mixed-integer programming models: General properties, *Journal of Optimization Theory and Applications* 16 (1976) 191–206.
- [8] J. P. Vielma, G. L. Nemhauser, Modeling disjunctive constraints with a logarithmic number of binary variables and constraints, *Mathematical Programming* 128 (2009) 49–72.
- [9] A. Martin, M. Möller, S. Moritz, Mixed Integer Models for the Stationary Case of Gas Network Optimization, *Mathematical Programming* 105 (2006) 563–582.
- [10] N. Alguacil, A. L. Motto, A. J. Conejo, Transmission expansion planning: a mixed-integer LP approach, *IEEE Transactions on Power Systems* 18 (2003) 1070–1077.
- [11] V. Gupta, I. E. Grossmann, An Efficient Multiperiod MINLP Model for Optimal Planning of Offshore Oil and Gas Field Infrastructure, *Industrial & Engineering Chemistry Research* 51 (2012) 6823–6840.
- [12] M. Carrion, J. M. Arroyo, A Computationally Efficient Mixed-Integer Linear Formulation for the Thermal Unit Commitment Problem, *IEEE Transactions on Power Systems* 21 (2006) 1371–1378.
- [13] L. Lodish, CALLPLAN: An interactive salesman’s call planning system, *Management Science* 18 (1971).
- [14] A. Borghetti, S. Member, C. D. Ambrosio, A. Lodi, S. Martello, An MILP Approach for Short-Term Hydro Scheduling and Unit Commitment With Head-Dependent Reservoir, *IEEE Transactions on Power Systems* 23 (2008) 1115–1124.
- [15] O. Günlük, J. Linderoth, Perspective reformulation and applications, *Mixed Integer Nonlinear Programming* (2012) 1–29.
- [16] M. W. Padberg, M. P. Rijal, Location, Scheduling, Design and Integer Programming, *Journal of The Operational Research Society* 49 (1998) 93–98.
- [17] K. L. Croxton, B. Gendron, T. L. Magnanti, Comparison of Mixed-Integer Programming Models for Nonconvex Piecewise Linear Cost Minimization Problems, *Management Science* 49 (2003) 1268–1273.
- [18] A. B. Keha, I. R. de Farias, G. L. Nemhauser, Models for representing piecewise linear cost functions, *Operations Research Letters* 32 (2004) 44–48.
- [19] O. Günlük, J. Linderoth, Perspective relaxation of mixed integer nonlinear programs with indicator variables, *Mathematical Programming Series B* 104 (2010) 186–203.
- [20] E. Balas, Disjunctive programming, in: *Annals of Discrete Mathematics 5: Discrete Optimization*, North Holland, 1979, pp. 3–51.
- [21] M. Padberg, Approximating separable nonlinear functions via mixed zero-one programs, *Operations Research Letters* 27 (2000) 1–5.
- [22] J. Lee, D. Wilson, Polyhedral methods for piecewise-linear functions i: the lambda method, *Discrete Applied Mathematics* 108 (2001) 269 – 285.

- [23] B. Geißler, A. Martin, A. Morsi, L. Schewe, Using piecewise linear functions for solving MINLPs, in: J. Lee, S. Leyffer (Eds.), *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, Springer, 2012, pp. 287–314.
- [24] A. Zoltners, P. Sinha, Integer programming models for sales resource allocation, *Management Science* 26 (1980) 242–260.

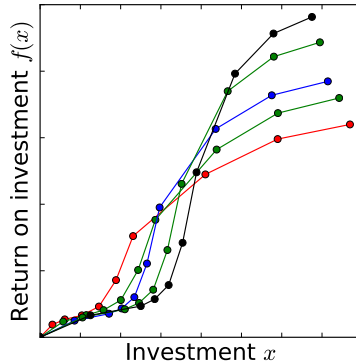
## Data generation

The three components of the instances used are the return on investment functions  $f_{jk}(\cdot) \forall j \in \mathcal{J} \ k \in \mathcal{K}$ , fixed costs  $G_j \forall j \in \mathcal{J}$  for entering each product market and variable costs  $c_{jk} \forall j \in \mathcal{J}, k \in \mathcal{K}$  per unit of budget allocated for marketing strategy  $k \in \mathcal{K}$  of product  $j \in \mathcal{J}$ . We now describe how each of these components were generated.

### Return on investment functions

In our sample application, the return on investment is evaluated by piecewise-linear functions  $f_{jk}(\cdot)$  which have the typical form shown in Figure 1.

Figure 2: Sample curves modeling return on investment for five different product/strategy pairs.



Let  $R(a, b, i, n)$  denote a random variable that lies between  $a + \frac{i(b-a)}{n}$  and  $a + \frac{(i+1)(b-a)}{n}$  with a distribution  $a + \frac{b-a}{n}(i + \beta(2, 2))$  where  $\beta(2, 2)$  is the beta distribution with both parameters set to 2. For each product  $j \in \mathcal{J}$ , the domain of  $f_{jk}(\cdot) \forall k \in \mathcal{K}$  was generated using

$$d_j \sim R(4, 8, j, |\mathcal{J}|)$$

and the range was generated as

$$r_j \sim R(0.5, 1, j, |\mathcal{J}|)$$

where the notation  $j$  overloads both the product  $j \in \mathcal{J}$  and an unique index for the product between 1 and  $\mathcal{J}$ . The desired s-shaped functions were generated by dividing the domain  $[0, d_j]$  of  $f_{jk}(\cdot) \forall k \in \mathcal{K}$  into three parts such that  $f_{jk}(\cdot)$  is concave increasing in  $[0, a_{jk}^1]$ , convex increasing in  $[a_{jk}^1, a_{jk}^2]$  and concave increasing again in  $[a_{jk}^2, d_j]$ . The random variables  $a_{jk}^2$  and  $a_{j,k}^2$  were generated using

$$a_{jk}^1 \sim d_j R(0.1, 0.5, j, |\mathcal{J}|)$$

$$a_{jk}^2 \sim d_j R(0.3, 0.7, j, |\mathcal{J}|).$$

The set of breakpoints  $B_{jki} \forall i \in \{1 \dots n\}$  were calculating by dividing each of the three domains into approximately  $\frac{n}{3}$  equal parts which can be written as

$$\begin{aligned} B_{jki} &= 3i \frac{a_{jk}^1}{n} & i &= 1 \dots \left\lfloor \frac{n}{3} \right\rfloor \\ B_{jki} &= a_{jk}^1 + 3i \frac{a_{jk}^2 - a_{jk}^1}{2n} & i &= \left\lfloor \frac{n}{3} \right\rfloor + 1 \dots \left\lfloor \frac{2n}{3} \right\rfloor \\ B_{jki} &= a_{jk}^2 + 3i \frac{d_j - a_{jk}^3}{n} & i &= \left\lfloor \frac{2n}{3} \right\rfloor + 1 \dots n. \end{aligned}$$

The corresponding function evaluations  $F_{jki} := f_{jk}(B_{jki})$  were generated as

$$\begin{aligned} F_{jki} &= b_{jk}^1 \sqrt{\frac{B_{jki}}{B_{jk\lfloor \frac{n}{3} \rfloor}}} & i &= 1 \dots \left\lfloor \frac{n}{3} \right\rfloor \\ F_{jki} &= F_{jk\lfloor \frac{n}{3} \rfloor} + b_{jk}^2 \left( \frac{B_{jki} - B_{jk\lfloor \frac{n}{3} \rfloor}}{B_{jk\lfloor \frac{2n}{3} \rfloor} - B_{jk\lfloor \frac{n}{3} \rfloor}} \right)^2 & i &= \left\lfloor \frac{n}{3} \right\rfloor + 1 \dots \left\lfloor \frac{2n}{3} \right\rfloor \\ F_{jki} &= F_{jk\lfloor \frac{2n}{3} \rfloor} + b_{jk}^3 \sqrt{\frac{B_{jki} - B_{jk\lfloor \frac{2n}{3} \rfloor}}{B_{jk\lfloor \frac{2n}{3} \rfloor} - B_{jk\lfloor \frac{n}{3} \rfloor}}} & i &= \left\lfloor \frac{2n}{3} \right\rfloor + 1 \dots n \end{aligned}$$

where  $b_{jk}^1$ ,  $b_{jk}^2$  and  $b_{jk}^3$  are random variables distributed by

$$\begin{aligned} b_{jk}^1 &\sim r_j R(0.05, 0.1, j, |\mathcal{J}|) \\ b_{jk}^2 &\sim r_j R(0.4, 0.7, j, |\mathcal{J}|) \\ b_{jk}^3 &\sim r_j R(0.7, 1, j, |\mathcal{J}|). \end{aligned}$$

#### Costs and Budget

For each strategy  $k \in \mathcal{K}$  and product  $j \in \mathcal{J}$ , the per-unit operating costs were generated as

$$c_{jk} \sim \beta(2, 2) R(0.8, 1.2, j, |\mathcal{J}|) R(0.8, 1.2, k, |\mathcal{K}|) \quad \forall j \in \mathcal{J}, k \in \mathcal{K}$$

and the fixed costs were generated as

$$G_j \sim E_G R(0.5, 1, j, |\mathcal{J}|) U(0.8, 1.2) \quad \forall j \in \mathcal{J}.$$

where  $E_G = 0.105|\mathcal{J}| |\mathcal{K}|$ . This procedure ensured that the total fixed costs are of the same order as the total variable costs. The overall budget  $D$  was set to  $6E_G$ .