Submitted to
manuscript

Mixed-Integer Models for Nonseparable Piecewise Linear Optimization: Unifying Framework and Extensions

Juan Pablo Vielma, Shabbir Ahmed and George Nemhauser
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332
{jvielma,sahmed,gnemhaus}@isye.gatech.edu

We study the modeling of non-convex piecewise linear functions as Mixed Integer Programming (MIP) problems. We review several new and existing MIP formulations for continuous piecewise linear functions with special attention paid to multivariate non-separable functions. We compare these formulations with respect to their theoretical properties and their relative computational performance. In addition, we study the extension of these formulations to lower semicontinuous piecewise linear functions.

Key words: Mixed Integer Programming, Piecewise Linear Functions

History: September, 2008

1. Introduction

We consider optimization problems involving piecewise linear functions modeled as Mixed Integer Programming (MIP) problems. When the functions considered are convex these problems can be modeled as Linear Programming (LP) problems, so we focus on non-convex functions for which the optimization problem is NP-hard even when all the functions are univariate (Keha et al. 2006).

Non-convex piecewise linear functions are generally used to approximate non-linearities arising from factors such as economies of scale or complex technological processes. They also naturally appear as cost functions of supply chain problems to model discounts for high volume and fixed charges. Applications of optimization problems with non-convex piecewise linear functions include production planning (Fourer et al. 1993), optimization of electronic circuits (Graf et al. 1990), operation planning of gas networks (Martin et al. 2006), process engineering (Bergamini et al. 2005, 2008), merge-in-transit (Croxton et al. 2003b) and other network flow problems with non-convex piecewise linear objective functions (Croxton et al. 2007).

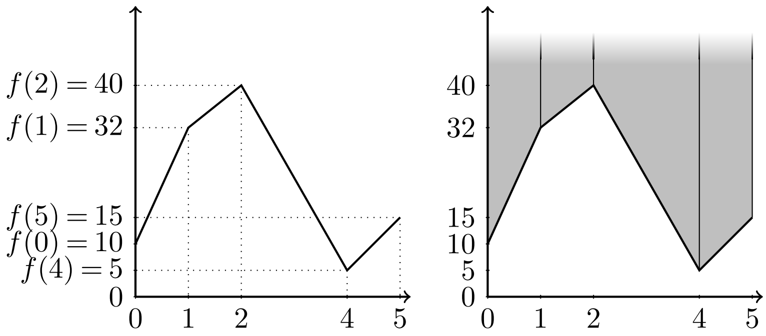
Optimization problems involving non-convex piecewise linear functions can be solved with specialized algorithms (de Farias Jr. et al. 2008, Keha et al. 2006, Tomlin 1981) or they can be modeled as MIPs (Lowe 1984, Sherali 2001, Croxton et al. 2003a, Balakrishnan and Graves 1989, Keha et al. 2004, Dantzig 1960, Wilson 1998, Lee and Wilson 2001, Jeroslow and Lowe 1985, Padberg 2000, Vielma and Nemhauser 2008a, Magnanti and Stratila 2004, Markowitz and Manne 1957) and solved with a general purpose MIP solver. The advantage of this latter approach is that it capitalizes on the advanced technology available in state of the art MIP solvers (Vielma et al. 2008). MIP models for non-convex piecewise linear functions have been extensively studied, but existing comparisons (Croxton et al. 2003a, Keha et al. 2004, Jeroslow and Lowe 1985) only concentrate on the case in which the functions are separable (i.e. can be written as the sum of univariate functions). When a non-separable function is known analytically it can sometimes be converted into a separable one by algebraic manipulations (Tomlin 1981). However this conversion might be undesirable for numerical reasons (Martin et al. 2006) and because it can result in weaker formulations (Croxton et al. 2007). Furthermore, in many applications the functions come from complicated simulation models (Lasdon and Waren 1980) and are not known analytically.

The main objective of this paper is to unify the numerous MIP models for piecewise linear functions into a common framework which considers the possibility of non-separable functions and discontinuities directly. In addition, we present a theoretical and computational comparison of the models considered. Because models for separable multivariate functions can be obtained directly from models for univariate functions we will assume that multivariate functions are non-separable.

The remainder of the paper is organized as follows. In Section 2 we study the MIP modeling of continuous piecewise linear functions and define concepts that will be used throughout the paper. In Section 3 we give several MIP models for continuous piecewise linear functions and in Section 4 we study some properties of these formulations. In Section 5, we present computational results comparing the formulations for continuous functions. In Section 6, we study the extension of the formulations to lower semicontinuous functions and in Section 7 we present computational results comparing the formulations for this class of functions. In Section 8 we present some final remarks.

2. Modeling Piecewise Linear Functions

An appropriate way of modeling a piecewise linear function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is to model its epigraph given by $\text{epi}(f) := \{(x, z) \in D \times \mathbb{R} : f(x) \leq z\}$. For example, the epigraph of the function in Figure 1(a) is depicted in Figure 1(b).



(a) f. (b) epi(f).

Figure 1 A continuous piecewise linear function and its epigraph as the union of polyhedra.

For simplicity, we assume that the function domain D is bounded and f is only used in a constraint of the form $f(x) \leq 0$ or as an objective function that is being minimized. We then need a model of epi(f) since $f(x) \leq 0$ can be modeled as $(x, z) \in \text{epi}(f)$, $z \leq 0$ and the minimization of f can be achieved by minimizing z subject to $(x, z) \in \text{epi}(f)$. For continuous functions we can also work with its graph, but modeling the epigraph will allow us to extend most of the results to some discontinuous functions and will simplify the analysis of formulation properties.

Following the theory developed by Jeroslow and Lowe (Jeroslow 1987, 1989, Jeroslow and Lowe 1984, 1985, Lowe 1984), we say that a polyhedron $P \subset \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^p \times \mathbb{R}^q$ is a binary mixed-integer programming model for a set $S \subset \mathbb{R}^n \times \mathbb{R}$ if

$$(x, z) \in S \Leftrightarrow \exists (\lambda, y) \in \mathbb{R}^p \times \{0, 1\}^q \text{ s.t. } (x, z, \lambda, y) \in P. \tag{1}$$

Under the bounded domain assumption, Jeroslow and Lowe prove that the epigraph of a function can be modeled as a binary mixed-integer programming model if and only if it is a union of polyhedra with a common recession cone given by $C_n^+ := \{(0, z) \in \mathbb{R}^n \times \mathbb{R} : z \geq 0\}$. This condition is a special case of the results in Jeroslow (1989), which also consider unbounded domains and more general uses of f in a mathematical program. Furthermore, this condition implies that for a function $f : D \subset \mathbb{R} \to \mathbb{R}$ we have that epi(f) can be modeled as a binary mixed-integer programming model if and only if f is piecewise linear and lower semicontinuous. Our definition of a piecewise linear function is motivated by the extension of this characterization to the multivariate case.

A single variable continuous piecewise linear function $f : [0, u] \to \mathbb{R}$ can be described as

$$f(x) := \Big\{ m_i x + c_i \quad x \in [d_{i-1}, d_i] \quad \forall i \in \{1, \ldots, K\} \tag{2}$$

for some $K \in \mathbb{Z}_+$, $\{m_i\}_{i=1}^K \subset \mathbb{R}$, $\{c_i\}_{i=1}^K \subset \mathbb{R}$ and $\{d_k\}_{k=0}^K \subset \mathbb{R}$ such that $0 = d_0 < d_1 < \ldots < d_K = u$. For example, function f depicted Figure 1(a) can be described in form (2) for $K = 4$, $m_1 = 22$, $m_2 = 8$, $m_3 = -17.5$, $m_4 = 10$, $c_1 = 10$, $c_2 = 24$, $c_3 = 75$, $c_4 = -35$, $d_0 = 0$, $d_1 = 1$, $d_2 = 2$, $d_3 = 4$ and $d_4 = 5$. A natural extension to the multivariate case is given by

Definition 1 (Continuous Piecewise Linear Function). Let $D \subset \mathbb{R}^n$ be a compact set. A continuous function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is a piecewise linear function if and only if there exists $\{m_P\}_{P \in \mathcal{P}} \subseteq \mathbb{R}^n$, $\{c_P\}_{P \in \mathcal{P}} \subseteq \mathbb{R}$ and a finite family of polytopes $\mathcal{P}$ such that $D = \bigcup_{P \in \mathcal{P}} P$ and

$$f(x) := \Big\{ m_P x + c_P \quad x \in P \quad \forall P \in \mathcal{P}. \tag{3}$$

Note that D does not need to be convex or connected and that the boundedness assumption is for simplicity. Furthermore, if $x \in P_1 \cap P_2$ for two polytopes $P_1, P_2 \in \mathcal{P}$ the definition implies that $m_{P_1} x + c_{P_1} = m_{P_2} x + c_{P_2}$ which ensures the continuity of f on D. In addition, Definition 1 does not specify how the polytopes are described as this is formulation dependent. In some formulations the polytopes are given as the convex hull of a finite number of points and in others the polytopes are given as a system of linear inequalities. The finite family of polytopes $\mathcal{P}$ is usually taken to be a triangulation of D (Lee and Wilson 2001, Martin et al. 2006, Wilson 1998) and in fact some models will require this. For any family of polytopes $\mathcal{P}$ we denote the set of vertices of the family by $\mathcal{V}(\mathcal{P}) := \bigcup_{P \in \mathcal{P}} V(P)$ where $V(P)$ is the set of vertices of P. When $\mathcal{P}$ is a triangulation this coincides with the usual definition of vertices of a triangulation.

Using the approach of modeling epi(f) as a union of polyhedra, Balas (Balas 1979) and Jeroslow and Lowe introduce two standard ways of modeling f. An advantage of this approach is that it allows for a simple treatment of lower semicontinuous functions. In addition, with this definition the epigraph of a continuous piecewise linear function is the union of polyhedra given by

$$\operatorname{epi}(f) = C_n^+ + \bigcup_{P\in\mathcal{P}} \operatorname{conv}\left(\{(v, f(v))\}_{v\in V(P)}\right) \tag{4}$$

where conv denotes the convex hull operation and + denotes the Minkowski addition of sets. For the function given in Figure 1(a) this characterization is illustrated in Figure 1(b) and detailed in Appendix EC.1.

3. Mixed Integer Programming Models for Piecewise Linear Functions

In this section we review several new and existing formulations for continuous functions. In Appendix EC.1 we illustrate the formulations for the function depicted in Figure 1(a).

3.1. Disaggregated convex combination models

All formulations in this section represent $(x,z) \in \operatorname{epi}(f)$ as the convex combination of points $(v, f(v))$ for $v \in \mathcal{V}(\mathcal{P})$ plus a ray in C_n^+. They have one continuous variable for each $v \in V(P)$ and for each $P \in \mathcal{P}$ to represent a point $(x,z) \in \operatorname{epi}(f)$ as $(x,z) = r + \sum_{P\in\mathcal{P}} \sum_{v\in V(P)} \lambda_{P,v}(v, f(v))$, for $r \in C_n^+$ and $\{\lambda_{P,v}\}_{P\in\mathcal{P},\, v\in V(P)} \subset \mathbb{R}_+$ such that $\sum_{P\in\mathcal{P}} \sum_{v\in V(P)} \lambda_{P,v} = 1$.

3.1.1. Basic Model

This formulation has no requirement on the family of polytopes and is given by

$$\sum_{P\in\mathcal{P}} \sum_{v\in V(P)} \lambda_{P,v} v = x, \qquad \sum_{P\in\mathcal{P}} \sum_{v\in V(P)} \lambda_{P,v}\left(m_P v + c_P\right) \le z \tag{5a}$$

$$\lambda_{P,v} \ge 0 \quad \forall P \in \mathcal{P},\, v \in V(P), \qquad \sum_{v\in V(P)} \lambda_{P,v} = y_P \quad \forall P \in \mathcal{P} \tag{5b}$$

$$\sum_{P\in\mathcal{P}} y_P = 1, \qquad y_P \in \{0,1\} \quad \forall P \in \mathcal{P}. \tag{5c}$$

This formulation has been studied in Croxton et al. (2003a), Jeroslow (1987), Jeroslow and Lowe (1984), Lowe (1984), Meyer (1976) and Sherali (2001) and is sometimes referred to as the *convex combination model.* To distinguish it from the formulation in Section 3.2 we instead refer to it as the *disaggregated convex combination model* and denote it by DCC.

3.1.2. Logarithmic Model

Using ideas from Ibaraki (1976), Vielma and Nemhauser (2008a) and Vielma and Nemhauser (2008b) we can reduce the number of binary variables and constraints of DCC. To do this we identify each polytope in $\mathcal{P}$ with a binary vector in $\{0,1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$ through an injective function $B : \mathcal{P} \to \{0,1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$. We then use $\lceil \log_2 |\mathcal{P}| \rceil$ binary variables $y \in \{0,1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$ to force $\sum_{v \in V(P)} \lambda_{P,v} = 1$ when $y = B(\mathcal{P})$.

The resulting formulation has no requirement on the family of polytopes and is given by

$$\sum_{P \in \mathcal{P}} \sum_{v \in V(P)} \lambda_{P,v} v = x, \qquad \sum_{P \in \mathcal{P}} \sum_{v \in V(P)} \lambda_{P,v} \left(m_P v + c_P\right) \leq z \quad \text{(6a)}$$

$$\lambda_{P,v} \geq 0 \quad \forall P \in \mathcal{P},\, v \in V(P), \qquad \sum_{P \in \mathcal{P}} \sum_{v \in V(P)} \lambda_{P,v} = 1 \quad \text{(6b)}$$

$$\sum_{P \in \mathcal{P}^+(B,l)} \sum_{v \in V(P)} \lambda_{P,v} \leq y_l, \quad \sum_{P \in \mathcal{P}^0(B,l)} \sum_{v \in V(P)} \lambda_{P,v} \leq (1 - y_l), \qquad y_l \in \{0,1\} \quad \forall l \in L(\mathcal{P}), \quad \text{(6c)}$$

where $B : \mathcal{P} \to \{0,1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$ is any injective function, $\mathcal{P}^+(B,l) := \{P \in \mathcal{P} : B(P)_l = 1\}$, $\mathcal{P}^0(B,l) := \{P \in \mathcal{P} : B(P)_l = 0\}$ and $L(\mathcal{P}) := \{1, \ldots, \lceil \log_2 |\mathcal{P}| \rceil\}$. We refer to it as the *logarithmic dissagregated convex combination model* and denote it by DLog.

3.2. Convex combination models

The formulations in this section reduce the number of continuous variables of DCC by aggregating variables associated with a point in $\mathcal{V}(\mathcal{P})$ that belongs to more than one polytope in $\mathcal{P}$. The resulting formulations have one continuous variable for each $v \in \mathcal{V}(\mathcal{P})$ and hence represent point $(x,z) \in \operatorname{epi}(f)$ as $(x,z) = r + \sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v (v, f(v))$, for $r \in C_n^+$ and $\lambda \in \mathbb{R}_+^{\mathcal{V}(\mathcal{P})}$ such that $\sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v = 1$.

3.2.1. Basic Model

This formulation has no requirement on the family of polytopes and is given by

$$\sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v v = x, \qquad \sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v \left(m_P v + c_P\right) \leq z \quad \text{(7a)}$$

$$\lambda_v \geq 0 \quad \forall v \in \mathcal{V}(\mathcal{P}), \qquad \sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v = 1 \quad \text{(7b)}$$

$$\lambda_v \leq \sum_{P \in \mathcal{P}(v)} y_P \quad \forall v \in \mathcal{V}(\mathcal{P}), \qquad \sum_{P \in \mathcal{P}} y_P = 1, \qquad y_P \in \{0,1\} \quad \forall P \in \mathcal{P}, \quad \text{(7c)}$$

where $\mathcal{P}(v) := \{P \in \mathcal{P} : v \in P\}$. This formulation is studied in Dantzig (1963, 1960), Garfinkel and Nemhauser (1972), Jeroslow and Lowe (1985), Keha et al. (2004), Lee and Wilson (2001), Lowe (1984), Nemhauser and Wolsey (1988), Padberg (2000) and Wilson (1998) and is sometimes referred to as the *lambda method*. We refer to this formulation as the *convex combination model* and denote it by CC.

3.2.2. Logarithmic Model

As in DLog's construction we can reduce the number of binary variables and constraints of CC by identifying each polytope in $\mathcal{P}$ with a binary vector in $\{0,1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$ through an injective function $B : \mathcal{P} \to \{0,1\}^{\lceil \log_2 |\mathcal{P}| \rceil}$. However, we now need B to comply with conditions that can be interpreted as the construction of a binary branching scheme for the effect of (7c) on $\lambda \in \mathbb{R}^{\mathcal{V}(\mathcal{P})}$. This constraint requires the non-zero λ variables to be associated with the vertices of a polytope in $\mathcal{P}$:

$$\exists P \in \mathcal{P} \text{ s.t. } \{v \in \mathcal{V}(\mathcal{P}) : \lambda_v > 0\} \subset V(P). \tag{8}$$

A binary branching scheme for (8) imposes it by fixing to zero disjoint sets of λ variables in each side of a series of branching dichotomies. For example, for the function depicted in Figure 1(a) we have $\mathcal{P} = \{[0,1],[1,2],[2,4],[4,5]\}$ and we can force (8) by the branching scheme given by the following two dichotomies: ($\lambda_2 = 0$ or $\lambda_0 = \lambda_5 = 0$) and ($\lambda_4 = \lambda_5 = 0$ or $\lambda_0 = \lambda_1 = 0$).

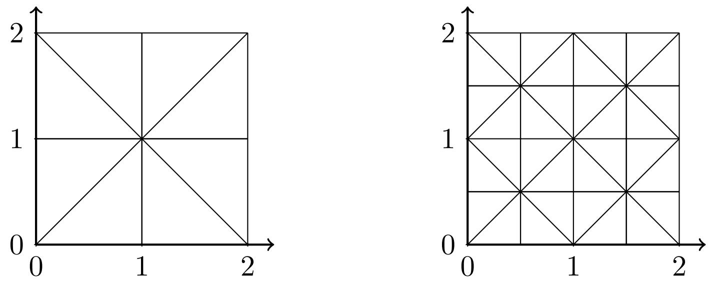
In general, a branching scheme for (8) is a family of dichotomies $\{L_s, R_s\}_{s\in S}$ indexed by a finite set S and with $L_s, R_s \subset \mathcal{V}(\mathcal{P})$ such that for every $P \in \mathcal{P}$ we have $V(P) = \bigcap_{s\in S}\big(\mathcal{V}(\mathcal{P}) \setminus T_s\big)$, where $T_s = L_s$ or $T_s = R_s$ for each $s \in S$. For such a branching scheme a valid formulation is given by

$$\sum_{v\in\mathcal{V}(\mathcal{P})} \lambda_v v = x, \qquad \sum_{v\in\mathcal{V}(\mathcal{P})} \lambda_v \left(m_P v + c_P\right) \le z \tag{9a}$$

$$\lambda_v \ge 0 \quad \forall v \in \mathcal{V}(\mathcal{P}), \qquad \sum_{v\in\mathcal{V}(\mathcal{P})} \lambda_v = 1 \tag{9b}$$

$$\sum_{v\in L_s} \lambda_v \le y_s, \qquad \sum_{v\in R_s} \lambda_v \le (1 - y_s), \qquad y_s \in \{0,1\} \quad \forall s \in \mathcal{S}. \tag{9c}$$

For (9) to have a logarithmic number of binary variables, we need a branching scheme with a logarithmic number of dichotomies. Such a scheme was introduced in Vielma and Nemhauser (2008a)



(a) J_1 triangulation of $[0,2]^2$. (b) 1/2 scaled J_1 triangulation of $[0,2]^2$.

Figure 2 Examples of triangulations of subsets of $\mathbb{R}^2$.

and Vielma and Nemhauser (2008b) for the case when the family of polytopes $\mathcal{P}$ is topologically equivalent or compatible (Aichholzer et al. 2003) with a triangulation known as J_1 or "Union Jack" (Todd 1977). For simplicity we first describe the formulation for the case when $\mathcal{P} = J_1$ and then show how to extend the formulation to the case where $\mathcal{P}$ is compatible with J_1.

J_1 is defined for $D = [0,K]^n$ for $K \in \mathbb{Z}$ even. The vertex set of J_1 is given by $\mathcal{V} = \{0,\dots,K\}^n$. The simplices of J_1 are constructed as follows. Let $N = \{1,\dots,n\}$, $\mathcal{V}^0 = \{v \in \mathcal{V} : v_i \text{ is odd}, \forall i \in N\}$, $\mathrm{Sym}(N)$ be the group of all permutations on N and e^i be the i-th unit vector of $\mathbb{R}^n$. For each $(v^0,\pi,s) \in \mathcal{V}^0 \times \mathrm{Sym}(N) \times \{-1,1\}^n$ define $j_1(v^0,\pi,s)$ to be the simplex whose vertices are $\{y^i\}_{i=0}^n$ where $y^0 = v^0$ and $y^i = y^{i-1} + s_{\pi(i)} e^{\pi(i)}$ for each $i \in N$. Triangulation J_1 of D is given by all these simplices, which is illustrated in Figure 2(a) for $D = [0,2]^2$. A branching scheme for J_1 is constructed by dividing index set S into two sets S_1 and S_2. The first set is given by $S_1 := N \times \{1,\dots,\lceil \log_2(K) \rceil\}$ and $L_{(s_1,s_2)} := \{v \in \mathcal{V} : v_{s_1} \in O(s_2,1)\}$, $R_{(s_1,s_2)} := \{v \in \mathcal{V} : v_{s_1} \in O(s_2,0)\}$ for each $(s_1,s_2) \in S_1$, where $O(l,b) := \left\{k \in \{0,\dots,K\} : (k = 0 \text{ or } G_l^k = b) \text{ and } \left(k = K \text{ or } G_l^{k+1} = b\right)\right\}$ for an arbitrary but fixed set of binary vectors $(G^l)_{l=1}^K \subset \{0,1\}^{\lceil \log_2(K) \rceil}$ such that G^l and G^{l+1} differ in at most one component for each $l \in \{1,\dots,\lceil \log_2(K) \rceil - 1\}$. There are many different sets of vectors with this property and they are usually referred to as *reflective binary* or *Gray codes* (Wilf. 1989). The second set is given by $S_2 := \{(s_1,s_2) \in N^2 : s_1 < s_2\}$ and $L_{(s_1,s_2)} := \{v \in \mathcal{V} : v_{s_1} \text{ is even and } v_{s_2} \text{ is odd}\}$, $R_{(s_1,s_2)} := \{v \in \mathcal{V} : v_{s_1} \text{ is odd and } v_{s_2} \text{ is even}\}$ for each $(s_1,s_2) \in S_2$.

Following Vielma and Nemhauser (2008a) and Vielma and Nemhauser (2008b) we refer to the formulation obtained with this scheme as the *logarithmic branching convex combination model* and denote it by Log. As mentioned before, Log can be extended to any family of polytopes $\mathcal{P}$ that is compatible with the J_1 triangulation. This requires the existence of a bijection $\varphi : \{0,\ldots,K\}^n \to \mathcal{V}(\mathcal{P})$ between the vertices of J_1 and the family $\mathcal{P}$ such that $v_1,\ldots,v_{n+1}$ are the vertices of a simplex in J_1 if and only if $\varphi(v_1),\ldots,\varphi(v_{n+1})$ are the vertices of a polytope in $\mathcal{P}$. For example, taking $\varphi : \{0,\ldots,4\}^2 \to \{0,1/2,1,3/2,2\}^2$ given by $\varphi(v_1,v_2) = (v_1/2, v_2/2)$ we have that the 1/2 scaled J_1 triangulation depicted in Figure 2(b) is compatible with the J_1 triangulation of $[0,4]^2$. Using bijection φ the formulation for $\mathcal{P}$ is simply obtained by replacing (9a) by $\sum_{v\in\mathcal{V}(\mathcal{P})} \lambda_v \varphi(v) = x$ and $\sum_{v\in\mathcal{V}(\mathcal{P})} \lambda_v \left(m_P \varphi(v) + c_P\right) \le z$.

A similar formulation can be obtained from a branching scheme introduced in Martin et al. (2006), but the resulting formulation has a linear instead of logarithmic number of binary variables.

3.3. Multiple choice model

This formulation has no requirement on the family of polytopes and is given by

$$\sum_{P\in\mathcal{P}} x^P = x, \qquad \sum_{P\in\mathcal{P}} \left(m_P x^P + c_P y_P\right) \le z \tag{10a}$$

$$A_P x^P \le y_P b_P \quad \forall P \in \mathcal{P} \tag{10b}$$

$$\sum_{P\in\mathcal{P}} y_P = 1, \qquad y_P \in \{0,1\} \quad \forall P \in \mathcal{P}, \tag{10c}$$

where $A_P x \le b_P$ is the set of linear inequalities describing P. This formulation has been studied in Balakrishnan and Graves (1989), Croxton et al. (2003a), Jeroslow and Lowe (1984) and Lowe (1984). We refer to this formulation as the *multiple choice model* and denote it by MC.

3.4. Incremental model

This formulation requires $\mathcal{P}$ to be a triangulation with a special ordering property. This property always holds for univariate functions so for simplicity we describe the formulation for this case first. For univariate function $f : [l,u] \to \mathbb{R}$ and for $\mathcal{P} = \{[d_{k-1}, d_k]\}_{k=1}^K$ where $l = d_0 \le d_1 \le \ldots \le d_K = u$, the formulation is given by

$$d_0 + \sum_{k=1}^K \delta_k \left(d_k - d_{k-1}\right) = x, \qquad f(d_0) + \sum_{k=1}^K \delta_k \left(f(d_k) - f(d_{k-1})\right) \le z \tag{11a}$$

$$\delta_1 \le 1, \quad \delta_K \ge 0, \quad \delta_{k+1} \le y_k \le \delta_k, \quad y_k \in \{0,1\} \quad \forall k \in \{1,\ldots,K-1\}. \tag{11b}$$

The extension to multivariate functions (Wilson 1998) requires the family of polytopes to be a triangulation $\mathcal{T}$ that complies with the following ordering properties:

O1. The simplices in $\mathcal{T}$ can be ordered as $T_1,\ldots,T_{|\mathcal{T}|}$ so that $T_i \cap T_{i-1} \neq \emptyset$ for $i \in \{2,\ldots,|\mathcal{T}|\}$.

O2. For the order above, the vertices of each simplex T_i can be ordered as $v_i^0,\ldots,v_i^{|V(T_i)|-1}$ in a way such that $v_{i-1}^{|V(T_i)|-1} = v_i^0$ for $i \in \{2,\ldots,|\mathcal{T}|\}$.

These properties are required to represent (x,z) *incrementally* akin to (11a) for the univariate case. Fortunately these conditions are met for many triangulations including J_1 (Wilson 1998).

For a given order complying with O1–O2 the formulation is given by

$$v_0^0 + \sum_{i=1}^{|\mathcal{T}|} \sum_{j=1}^{|V(T_i)|-1} \delta_i^j \left(v_i^j - v_i^0\right) = x, \qquad f(v_0^0) + \sum_{i=1}^{|\mathcal{T}|} \sum_{j=1}^{|V(T_i)|-1} \delta_i^j \left(f(v_i^j) - f(v_i^0)\right) \leq z \tag{12a}$$

$$\sum_{j=1}^{|V(T_1)|-1} \delta_1^j \leq 1, \qquad \delta_i^j \geq 0 \quad \forall i \in \{1,\ldots,|\mathcal{T}|\},\, j \in \{1,\ldots,|V(T_i)|-1\} \tag{12b}$$

$$y_i \leq \delta_i^{|V(T_i)|-1}, \qquad \sum_{j=1}^{|V(T_{i+1})|-1} \delta_{i+1}^j \leq y_i, \qquad y_i \in \{0,1\} \quad \forall i \in \{1,\ldots,|\mathcal{T}|-1\}. \tag{12c}$$

This formulation has been studied in Croxton et al. (2003a), Dantzig (1963, 1960), Keha et al. (2004), Markowitz and Manne (1957), Padberg (2000), Sherali (2001), Vajda (1964) and Wilson (1998) and it is sometimes referred to as the *delta method*. Following Croxton et al. (2003a) and Keha et al. (2004) we refer to it as the *incremental model* and denote it by Inc.

4. Properties of Mixed Integer Programming Formulations

In this section we study some properties of the formulations. We begin by studying the strength of the formulations as a model of epi(f) ignoring possible interactions with other constraints. For this case a motivating problem is the minimization of $f: D \subset \mathbb{R}^n \to \mathbb{R}$ over its domain D given by

$$\min_{x \in D} f(x) = \min_{(x,z) \in \mathrm{epi}(f)} z. \tag{13}$$

We then study the effects of interactions with other constraints using as a motivating problem

$$\min_{x \in X} f(x) = \min_{(x,z) \in \mathrm{epi}(f) \cap (X \times \mathbb{R})} z, \tag{14}$$

where $X \subset D$ is any compact set. Finally, we study the sizes of the formulations and their requirements on the family of polytopes $\mathcal{P}$ used to describe the piecewise linear function.

Consider a MIP formulation of epi(f) given by a polytope $P \subset \mathbb{R}^{n+p+q+1}$ complying with (1). The linear programming (LP) relaxation of the formulation is then simply P. Alternative MIP formulations are usually compared with respect to the tightness of their LP relaxation in the absence of additional constraints. In this regard, the strongest possible property of a MIP formulation is to require that all vertices of its LP relaxation comply with the corresponding integrality requirements. Formulations with this property are referred to as *locally ideal* in Padberg (2000) and Padberg and Rijal (1996). It is shown in Lee and Wilson (2001), Padberg (2000) and Wilson (1998) that CC is not locally ideal. However all of the other formulations from Section 3 are locally ideal.

THEOREM 1. *All formulations from Section 3 except CC are locally ideal.*

The proof of this and other statements in this section are given in Appendix EC.2. For a locally ideal formulation P of epi(f) we have

$$\min_{(x,z,\lambda,y)\in P} z = \min_{x\in D} f(x), \tag{15}$$

which allows solving (13) directly as an LP and can be useful for solving (14) with a branch-and-bound algorithm. However, as noted in Croxton et al. (2003a) and Keha et al. (2004), property (15) might still hold for non-locally ideal formulations such as CC. In fact, we will see that (15) is implied by a geometric property introduced by Jeroslow and Lowe, but is weaker than the locally ideal property.

A slightly restricted version of Proposition 3.1 in Jeroslow and Lowe (1984) states that for any closed set $S \subset \mathbb{R}^n \times \mathbb{R}$ and for any binary mixed-integer programming model $P \subset \mathbb{R}^{n+p+q+1}$ for S, the projection of P onto the first $n+1$ variables contains the convex hull of S. Jeroslow and Lowe referred to a model P of S as *sharp* when the projection is exactly the convex hull of S. By letting S be the epigraph of piecewise linear function f we directly get the following result.

Theorem 2. *(Croxton et al. 2003a, Jeroslow and Lowe 1984, Lowe 1984) Let* $D \subset \mathbb{R}^n$ *be a polytope,* $f : D \to \mathbb{R}$ *be a continuous piecewise linear function,* $P \subset \mathbb{R}^{n+p+q+1}$ *be a MIP formulation for* $\mathrm{epi}(f)$ *satisfying* (1) *and* $P_{(x,z)}$ *the projection of* P *onto* (x,z)*. Then* $\mathrm{epi}(\mathrm{convenv}_D(f)) = \mathrm{conv}(\mathrm{epi}(f)) \subset P_{(x,y)}$ *where* $\mathrm{convenv}_D$ *is the lower convex envelope of* f *over* D*.*

A formulation P of $\mathrm{epi}(f)$ is said to be sharp when $\mathrm{epi}(\mathrm{convenv}_D(f)) = \mathrm{conv}(\mathrm{epi}(f)) = P_{(x,y)}$. Because $\min_{x\in D} f(x) = \min_{x\in D} \mathrm{convenv}_D(f)(x)$ we have that (15) holds for sharp formulations. Sharpness has been shown to hold for some formulations in Croxton et al. (2003a), Jeroslow (1987, 1989), Jeroslow and Lowe (1984, 1985), Keha et al. (2004), Lowe (1984), Padberg (2000) and Sherali (2001) and the following proposition states that it holds for any locally ideal formulation.

Proposition 1. *Any locally ideal formulation is sharp.*

We then directly have that all formulations except CC are sharp. As noted in Section 3.2, CC can be obtained from DCC in a way which reduces its tightness. Fortunately, this loss of tightness does not affect the sharpness properties of CC so the following theorem holds.

Theorem 3. *All formulations from Section 3 are sharp.*

Sharpness is not preserved when x complies with additional constraints, so a property similar to (15) does not hold for (14). However, it is still possible to characterize the LP bound obtained when a sharp formulation is used to model the objective function of a larger model. The following theorem follows directly from the definitions of sharpness and convex envelopes.

Theorem 4. *Let* $D \subset \mathbb{R}^n$ *be a polytope,* $f : D \to \mathbb{R}$ *be a continuous piecewise linear function,* $P \subset \mathbb{R}^{n+p+q+1}$ *be a sharp binary mixed-integer programming model for* $\mathrm{epi}(f)$ *and* X *be a compact set. Then* $\min_{x,z,\lambda,y}\{z : (x,z,\lambda,y) \in P, \quad x \in X\} = \min_{x\in X} \mathrm{convenv}_D(f)(x)$.

For the case where X is a polytope this has also been studied in Croxton et al. (2003a) and Croxton et al. (2007) and together with Theorem 3 yields the following corollary.

Corollary 1. *All formulations from Section 3 give the same LP bound for solving* (14)*.*

Now we present the sizes of all the formulations given in Section 3. We give the number of extra constraints and extra variables besides z and x and also indicate the number of extra variables that are binary. Table 1 shows this information for all models. Except for Log and MC the sizes are given as a function of n, $|\mathcal{P}|$ and the number of vertices $|\mathcal{V}(\mathcal{P})|$ or $|V(P)|$. For MC the size is a function of n, $|\mathcal{P}|$ and the number of facets of polytope P denoted by $F(P)$. In particular if $\mathcal{P}$ is a triangulation we have that $|F(P)| \leq n+1$ for all $P \in \mathcal{P}$. For Log the size is a function of $|\mathcal{V}(\mathcal{P})|$ and $|S|$ where S is the branching scheme for the J_1 triangulation of $[0,K]^n$. In this case we have $|\mathcal{P}| = K^n n!$ and $|S| = n\lceil \log_2(K) \rceil + n(n-1)/2$, but it is not clear how to explicitly relate these numbers together when $n > 2$. However we can see that $|S|$ grows asymptotically as $\log_2(|\mathcal{P}|)$ only when n is fixed. More specifically, for fixed n we have $|S| \sim \log_2(|\mathcal{P}|)$ (i.e. $\lim_{K\to\infty} |S|/\log_2(|\mathcal{P}|) = 1$) with $|S| = \log_2(|\mathcal{P}|)$ for K of the form 2^r, but for fixed K we have $\log_2(|\mathcal{P}|) \in o(|S|)$ (i.e. $\lim_{n\to\infty} \log_2(|\mathcal{P}|)/|S| = 0$).

Model	Constraints	Additional Variables	Binaries
DCC	$n+\|\mathcal{P}\|+2$	$\|\mathcal{P}\| + \sum_{P\in\mathcal{P}} \|V(P)\|$	$\|\mathcal{P}\|$
DLog	$n+2\lceil\log_2(\|\mathcal{P}\|)\rceil+2$	$2\lceil\log_2(\|\mathcal{P}\|)\rceil + \sum_{P\in\mathcal{P}} \|V(P)\|$	$2\lceil\log_2(\|\mathcal{P}\|)\rceil$
CC	$n+3+\|\mathcal{V}(\mathcal{P})\|$	$\|\mathcal{V}(\mathcal{P})\|+\|\mathcal{P}\|$	$\|\mathcal{P}\|$
Log	$n+2+2\|S\|$	$\|\mathcal{V}(\mathcal{P})\|+\|S\|$	$\|S\|$
MC	$n+2+\sum_{P\in\mathcal{P}} F(P)$	$(n+1)\|\mathcal{P}\|$	$\|\mathcal{P}\|$
Inc	$1+2\|\mathcal{P}\|$	$\|\mathcal{P}\|-1+\sum_{P\in\mathcal{P}}(\|V(P)\|-1)$	$\|\mathcal{P}\|-1$

Table 1 Sizes of Formulations

Finally, we summarize the requirements that the different formulations have on the family of polytopes $\mathcal{P}$ used to describe the piecewise linear function. The first type of requirement concerns the description of the polytopes in $\mathcal{P}$ as either the convex hull of a finite number of points (vertex representation) or as the feasible region of a system of linear inequalities (inequality representation). Although conversion between the two descriptions can be done efficiently for special cases of $\mathcal{P}$ such as triangulations, the description requirements can be an important factor in the choice of the formulation when general polytopes are used. We have seen that every formulation except MC uses the vertex representation. The second type of requirements concerns the need for a particular family of polytopes $\mathcal{P}$. Although requiring $\mathcal{P}$ to be of a special class such as a triangulation is usually not too restrictive, it can be an important factor when the function is constructed as the interpolation of a non-linear function (Carnicer and Floater 1996, Pottmann et al. 2000). We have seen that DCC, DLog, CC, and MC have no requirement on $\mathcal{P}$. Inc requires $\mathcal{P}$ to be any triangulation which complies with conditions O1–O2 described in Section 3.4 and Log requires $\mathcal{P}$ to be the J_1 triangulation.

5. Computational Experiments for Continuous Functions

In this section we computationally test the formulations for continuous piecewise linear functions. Our tests are on transportation problems with piecewise linear objective functions. We believe these problems provide enough additional constraints to provide meaningful results while allowing the piecewise linear objectives to dominate the optimization effort.

All models were generated using Ilog Concert 2 and solved using CPLEX 11 on a 2.4GHz workstation with 2GB of RAM. Furthermore, all tests were run with a time limit of 10000 seconds.

5.1. Continuous Separable Concave Functions

The first set of experiments considers formulations for univariate functions. The instances tested for these formulations are the same transportation problems with concave separable piecewise linear objectives considered in Vielma and Nemhauser (2008a). These instances are based on the 10×10 transportation problems used in Keha et al. (2006) and Vielma et al. (2008). Each of the problems include the supply and demand information and capacities u_e for each arc e. The problems also include the subdivision of $[0, u_e]$ into 4 randomly selected intervals and their generation is described in Keha et al. (2006). For each of the 5 instances we constructed several randomly generated piecewise linear separable objective functions. These objective functions are of the form $\sum_{e \in E} f_e(x_e)$ where E is the set of arcs of the transportation problem and $f_e(x_e)$ is a continuous non-decreasing concave piecewise linear function of the flow x_e on arc e. We chose to use this class of functions because they are widely used in practice and are challenging enough to provide meaningful computational results. Each $f_e(x_e)$ is affine in K segments and has $f_e(0) = 0$. The slopes for each segment of a particular f_e were generated by obtaining a sample of size K from set $\{z/1000 : z \in \{1,\ldots,2000\}\}$ and sorting them to ensure concavity. We considered $K = 4$, 8, 16, and 32 and for each K and for each of the 5 transportation problems we generated 20 objective functions for a total of 100 instances for each K. To obtain the subdivisions of $[0, u_e]$ into 8, 16 and 32 intervals we simply recursively divided in half each of the intervals starting with the original 4 from Keha et al. (2006). Furthermore, we independently generated the objective functions for each choice of K.

We tested all mixed integer formulations from Section 3 and in addition we tested the traditional SOS2 formulation of piecewise linear functions (see for example Keha et al. (2004)) which does not include binary variables. We implemented this formulation using CPLEX's built in support for SOS2 constraints and we refer to it as SOS2 in the computational results.

Table 2 shows the minimum, average, maximum and standard deviation of the solve times in seconds. The table also shows the number of times the solves failed because the time limit was reached and the number of times each formulation had the fastest solve time (win or tie).

model	min	**avg**	max	std	win	fail
MC	0.2	**1.3**	8.3	1.5	45	0
SOS2	0.1	**1.7**	7.9	1.3	26	0
Log	0.2	**2.1**	12.4	2.3	24	0
DLog	0.3	**2.1**	10.3	2.2	4	0
Inc	0.4	**2.4**	11.6	2.5	2	0
DCC	0.3	**2.6**	14.0	2.5	0	0
CC	0.3	**4.6**	23.0	4.3	0	0

(a) 4 segments.

model	min	**avg**	max	std	win	fail
MC	1.2	**9.9**	39	7.0	41	0
Log	0.6	**12.3**	84	10.5	31	0
DLog	0.8	**13.2**	91	11.6	5	0
SOS2	0.8	**15.8**	202	23.0	23	0
DCC	2.6	**42.7**	252	46.6	0	0
Inc	5.1	**43.0**	163	29.3	0	0
CC	2.6	**81.0**	570	96.6	0	0

(b) 8 segments.

model	min	**avg**	max	std	win	fail
Log	0.5	**24**	96	18	80	0
DLog	0.8	**32**	132	25	17	0
MC	1.9	**97**	730	122	2	0
SOS2	1.9	**109**	1030	167	1	0
Inc	29.8	**302**	1442	239	0	0
CC	3.9	**351**	3691	517	0	0
DCC	3.9	**1366**	10000	2120	0	3

(c) 16 segments.

model	min	**avg**	max	std	win	fail
Log	2.5	**43**	194	39	90	0
DLog	5.5	**63**	328	53	8	0
SOS2	10.0	**925**	10000	1900	2	2
Inc	271.0	**981**	4039	685	0	0
CC	67.5	**1938**	10000	2560	0	4
MC	22.5	**2246**	10000	3208	0	9
DCC	89.6	**8163**	10000	3141	0	69

(d) 32 segments.

Table 2 Solve times for univariate continuous functions [s].

For $K = 4$ we see that the average solve time for all formulations is of the same order of magnitude, but for larger K's the difference between models becomes noticeable. Many conclusions could be extracted from these results, but they should be taken with care as they can depend on both the instances and the solver used. For example, MC is faster on average than Inc for K's ranging from 4 to 16, but in previous tests using CPLEX 9.1 the average solve time for Inc was always better than or comparable to MC. Nevertheless, we make the following observations.

We see that the logarithmic formulations Log and DLog can have a significant advantage over the other formulations (up to over an order of magnitude for $K = 32$) for K's larger than 4 and that, as expected, this advantage grows with K. Another interesting observation concerns SOS2, which in previous tests with CPLEX 9.1 was significantly slower than most mixed integer programming formulations. It seems that the reason for this bad performance was more of an implementation issue than a property of the SOS2 based formulation (Vielma et al. 2008). As the results show, the implementation of SOS2 constraints has been significantly improved in CPLEX 11 which allows SOS2 to always be among the 5 best formulations. In fact, it is only for $K = 32$ that we have mixed integer formulations outperforming SOS2 by more than an order of magnitude.

In an attempt to explain the results from Table 2 we study some characteristics of the solves by CPLEX. In Table 3 we present some results for the instances with $K = 8$ (Appendix EC.3 includes the same results for $K = 4$, $K = 16$, and $K = 32$). Corollary 1 states that all MIP formulations should provide the same LP relaxation bound and so should SOS2 (Keha et al. 2004). We confirmed this is true up to small numerical errors and, as expected, the common bound was not equal to the optimal MIP solution resulting in an average integrality GAP of 5% (calculated as $100(z_{IP} - z_{LP})/z_{IP}$ where z_{IP} and z_{LP} are the optimal values of the mixed integer program and its LP relaxation respectively). However, an equality in the LP relaxation bound does not necessarily imply an equality on the LP bound obtained at the root node by CPLEX as this includes preprocessing and cuts. For this reason we present in Table 3(a) the percentage of the integrality GAP that was closed by CPLEX at the root node for the different formulations (this was calculated as $100(z_{root} - z_{LP})/(z_{IP} - z_{LP})$ where z_{root} is the optimal values of the root relaxation obtained by CPLEX after preprocessing and cutting planes). A second issue is the time required to solve the LP relaxation of the different formulations, which we present in Table 3(b). Because the solve times were very small for all formulations, the results from Table 3(b) are in milliseconds. Finally, in Table 3(c) we present the number of nodes processed by CPLEX. The complexity of CPLEX makes it hard to infer categorical conclusions about these results, but we will comment on some interesting patterns. Note that a larger formulation might have an LP relaxation which is slower to solve, but it might allow CPLEX to close a larger percentage of the integrality GAP. This can lead to fewer branch-and-bound nodes needed to solve the problem, which can translate to faster solve speeds. An example of this behavior is MC, which has the second slowest solve time for its LP relaxation, but allows CPLEX to close the largest percentage of the integrality GAP resulting in the best performances in both number of nodes and solve times. On the other hand, having a small formulation can have the reverse effect on the LP relaxation solve speeds and closed GAP, but might still provide an advantage. For example, SOS2 is one of the smallest formulations as it does not include any binary variables. We can see that CPLEX does not close a significant percentage of the integrality GAP for SOS2, which translates into a need to process a large number of nodes. However, having the fastest solve time for its LP relaxation allows this formulation to still have an excellent performance with respect to solve times. Still, faster solves of its LP relaxation and

model	min	**avg**	max	std
MC	17.4	**37**	61.2	9.5
DCC	11.8	**23**	38.6	6.0
DLog	4.7	**20**	46.7	7.2
Log	6.2	**19**	35.6	6.6
Inc	5.3	**19**	39.3	6.9
CC	9.3	**16**	39.2	4.9
SOS2	0.0	**0**	1.2	0.1

(a) GAP closed at root node by CPLEX. [%]

model	min	**avg**	max	std
SOS2	0	**4.3**	10	4.3
Log	0	**9.5**	20	9.5
DCC	0	**11.5**	20	11.5
CC	0	**11.6**	20	11.6
DLog	0	**11.9**	20	11.9
MC	10	**24.3**	40	24.3
Inc	20	**38.1**	50	38.1

(b) LP relaxation solve time. [ms]

model	min	**avg**	max	std
MC	64	**535**	2003	301
Inc	142	**1970**	8814	1611
DLog	134	**2419**	17114	2415
Log	120	**2591**	22541	2777
DCC	549	**13956**	120035	19253
CC	500	**17276**	127110	22467
SOS2	606	**21833**	337199	39081

(c) Branch-and-bound nodes processed.

Table 3 Solve characteristics for univariate continuous functions and $K = 8$.

large percentages of root GAP closed might not necessarily translate to better performance. For example, DCC is on average better than Inc and DLog with respect to both solve speed of its LP relaxation and GAP closed at the root node. However, Inc and DLog have a better or comparable performance than DCC in terms of both solve times and nodes processed. This is particularly surprising for DLog which is essentially the same as DCC but with fewer variables. A possible explanation for this behavior is that Log, Inc and DLog allow CPLEX to perform a more effective branch-and-bound search. DCC produces unbalanced branch-and-bound trees as fixing a binary variable to zero produces very little change compared to fixing the same variable to one. In contrast, Log and DLog are designed to produce balanced branch-and-bound trees, and Inc also produces a fairly balanced tree since fixing a binary to a particular value in Inc usually fixes many other variables to take the same value.

5.2. Continuous Non-Separable Functions

We now consider non-separable functions of two variables. For these experiments we selected a series of two commodity transportation problems with 5 supply nodes and 2 demand nodes. These instances were constructed by combining two 5×2 transportation problems generated in a manner similar to the instances used in Vielma et al. (2008). The supplies, demands and individual commodity arc capacities for each commodity were obtained from two different transportation problems and the joint arc capacities were set to 3/4 of the sum of the corresponding individual arc capacities. We considered an objective function of the form $\sum_{e\in E} f_e(x_e^1, x_e^2)$ where E is the common set of 10 arcs of the transportation problems and $f_e(x_e^1, x_e^2)$ is a piecewise linear function of the flows x_e^i in arc e of commodity i for $i = 1, 2$. Each $f_e(x_e^1, x_e^2)$ for arc e with individual arc capacities u_e^i for commodity $i = 1, 2$ was constructed by triangulating $[0, u_e^1] \times [0, u_e^2]$ with the J_1 triangulation induced by the grid obtained from the subdivision of $[0, u_e^1]$ and $[0, u_e^2]$ into K intervals as determined from the respective original transportation problems. For K ranging from 4 to 16 the number of vertices and triangles range from 25 to 289 and from 32 to 512 respectively. Using this triangulation we then obtained $f_e(x_e^1, x_e^2)$ by interpolating $g\left(\|(x_e^1, x_e^2)\|\right)$ where $\|\cdot\|$ is the Euclidean norm and $g : [0, \|(u_e^1, u_e^2)\|] \to \mathbb{R}$ is a continuous concave piecewise linear function randomly generated in a similar way to the univariate functions of Section 5.1. The idea of this function is to use the sub-linearity of the Euclidean norm to consider discounts for sending the two commodities in the same arc and concave function g to consider economies of scale. We selected 5 combinations of different pairs of the original transportation problems and for each one of these we generated 20 objective functions for a total of 100 instances for each K.

Table 4 shows the usual statistics for the solve times with different grid sizes for all the appropriate formulations. We again used a limit of 10000 seconds and only tested a formulation for the next largest K if it had failed in less than 5 instances in the previous K.

model	min	**avg**	max	std	wins	fail
Log	0.4	**2.7**	9.3	2.0	93	0
MC	1.2	**5.6**	17.1	3.1	7	0
DLog	1.6	**7.6**	25.5	5.2	0	0
CC	5.9	**17.8**	107.2	14.5	0	0
Inc	2.8	**31.7**	126.5	25.8	0	0
DCC	8.1	**36.8**	476.1	50.6	0	0

(a) 4×4 grid.

model	min	**avg**	max	std	wins	fail
Log	1.7	**13**	33	5.4	100	0
DLog	17.8	**45**	135	20.2	0	0
MC	30.9	**398**	5328	583.6	0	0
Inc	99.5	**769**	6543	1110.5	0	0
CC	102.9	**4412**	10000	3554.6	0	13
DCC	237.0	**6176**	10000	3385.9	0	31

(b) 8×8 grid.

model	min	**avg**	max	std	wins	fail
Log	27	**56**	118	19	100	0
DLog	125	**325**	1064	128	0	0
Inc	772	**4857**	10000	3429	0	20
MC	2853	**9266**	10000	1678	0	77

(c) 16×16 grid.

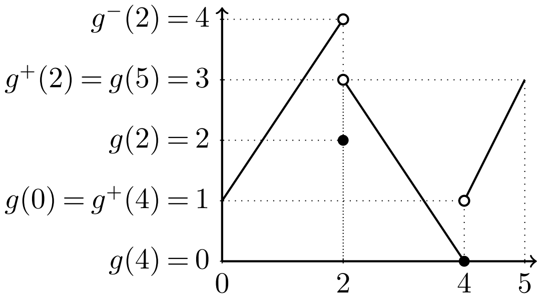
Table 4 Solve times for two variable multi-commodity transportation problems. [s].

Logarithmic models Log and DLog were among the best performers for all grid sizes, probably because for two variable functions $|\mathcal{P}|$ grows much faster with k than in the univariate case. For example, for $k = 4$ a $k \times k$ grid yields $|\mathcal{P}| = 32$ which is comparable to $k = 32$ in the univariate case. In addition, the smaller number of continuous variables is what probably allows Log to be the best performer overall.

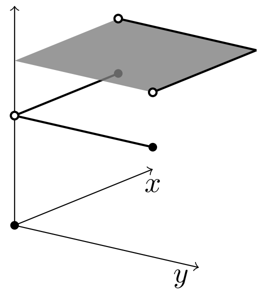
6. Extension to Lower Semicontinuous Functions

In this section we study the extension of the formulations to discontinuous functions such as the ones in Figure 3. Consider first the univariate piecewise linear discontinuous function g depicted in

Figure 3(a), for which $g^-(d) = \lim_{\substack{x\to d\\ x\le d}} g(x)$ and $g^+(d) = \lim_{\substack{x\to d\\ x\ge d}} g(x)$. Function g is now only affine in $[0,2)$, $\{2\}$, $(2,4]$ and $(4,5]$. However, because g is lower semicontinuous we have that epi(g) is closed and is still the union of polyhedra with common recession cone C_1^+. Hence we can model epi(g) as a binary mixed-integer programming problem. The example from Figure 3(a) shows that



(a) g



(b) h

Figure 3 Lower semicontinuous piecewise linear functions.

to consider discontinuous univariate piecewise linear functions we need to use intervals that are not necessarily of the form $[d_{i-1}, d_i]$ for $d_{i-1} < d_i$. The inclusion of points described as $\{d\} = [d, d]$ complies with Definition 1 as we did not require the polytopes to be full dimensional. In contrast, the inclusion of non closed intervals such as $[0,2)$ requires the use of sets other than polytopes. The simplest extension we can use is to consider bounded sets that can be described by a finite number of strict and non-strict linear inequalities. These sets are usually referred to as *copolytopes* (Kannan 1992). Using copolytopes instead of polytopes we get the following definition for not necessarily continuous piecewise linear functions.

Definition 2 (Piecewise Linear Function). Let $D \subset \mathbb{R}^n$ be a compact set. A (not necessarily continuous) function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is piecewise linear if and only if there exists a finite family of copolytopes $\mathcal{P}$ complying with $D = \bigcup_{P\in\mathcal{P}} P$ and (3) for some $\{m_P\}_{P\in\mathcal{P}} \subseteq \mathbb{R}^n$ and $\{c_P\}_{P\in\mathcal{P}} \subseteq \mathbb{R}$.

For example, function h from Figure 3(b) can be described as

$$h(x,y) := \begin{cases} 3 & (x,y) \in P_1 \\ 2 & (x,y) \in P_2 \\ 2 & (x,y) \in P_3 \\ 0 & (x,y) \in P_4. \end{cases} \qquad (16)$$

for $P_1 = (0,1]^2$, $P_2 = \{(x,y) \in \mathbb{R}^2 : x = 0, y > 0\}$, $P_3 = \{(x,y) \in \mathbb{R}^2 : y = 0, x > 0\}$ and $P_4 = \{(0,0)\}$.

A piecewise linear function as defined in Definition 2 is not necessarily lower semicontinuous, but this condition is crucial for obtaining a mixed integer programming model. For a lower semicontinuous piecewise linear function f we have a direct extension of characterization (4) to

$$\text{epi}(f) = C_n^+ + \bigcup_{P \in \mathcal{P}} \text{conv}\left(\{(v, m_P v + c_P)\}_{v \in V(\overline{P})}\right), \tag{17}$$

where $V(\overline{P})$ denotes the set of vertices of the closure $\overline{P}$ of P. We note that the closure of a copolytope $P = \{x \in \mathbb{R}^n : a_i x \leq b_i \, \forall i \in \{1,\ldots,p\}, a_i x < b_i \, \forall i \in \{p+1,\ldots,m\}\}$ is $\overline{P} = \{x \in \mathbb{R}^n : a_i x \leq b_i \, \forall i \in \{1,\ldots,m\}\}$. We now study the extension of formulations from Section 3 to the lower semicontinuous case and comment on the properties of the extensions.

Formulations DCC, DLog and MC directly model epi(f) so their extension to the lower semicontinuous case is achieved by replacing characterization (4) of epi(f) for continuous f by characterization (17) of epi(f) for lower semicontinuous f. Because $V(P)$ in (4) is replaced by $V(\overline{P})$ in (17) the extension of DCC is obtained by replacing $V(P)$ by $V(\overline{P})$ in (5). For univariate functions this extension has been noted in Croxton et al. (2003a) and Sherali (2001). Similarly, the extension of DLog is obtained by replacing $V(P)$ by $V(\overline{P})$ in (6). The extension of MC is obtained from (10) by replacing (10b) by $A_{\overline{P}} \lambda_P \leq y_P b_{\overline{P}} \quad \forall P \in \mathcal{P}$ where $A_{\overline{P}} \lambda_P \leq b_{\overline{P}}$ is the set of linear inequalities describing polytope $\overline{P}$. For univariate functions this extension has been noted in Croxton et al. (2003a). In Appendix EC.4 we illustrate these formulations for the function depicted in Figure 3(a).

Additionally, for simple discontinuities we can use ad-hoc techniques to adapt other formulations as well. We explore two such techniques for univariate functions.

The first technique is from Vielma et al. (2008) and involves duplicating break points at which a univariate function is discontinuous. For a univariate lower semicontinuous piecewise linear function $f : [0,u] \to \mathbb{R}$ we always have an integer K and real numbers $(d_k)_{k=0}^K$ and $(f_k)_{k=0}^K$ such that $0 = d_0 \leq d_1 \leq \ldots \leq d_K = u$, f_k is equal to $f(d_k)$, $f^-(d_k)$ or $f^+(d_k)$ and $\text{epi}(f) = C_1^+ + \left(\bigcup_{k=1}^K \text{conv}\left(\{(d_{k-1}, f_{k-1}), (d_k, f_k)\}\right)\right)$. Using this characterization we can adapt CC to obtain the formulation given by

$$\sum_{k=0}^{K} \lambda_k d_k = x, \quad \sum_{k=0}^{K} \lambda_k f_k \leq z, \quad \sum_{k=0}^{K} \lambda_k = 1, \quad \lambda_k \geq 0 \quad \forall k \in \{0, \ldots, K\} \tag{18a}$$

$$\lambda_0 \leq y_1, \quad \lambda_K \leq y_K, \quad \lambda_k \leq y_k + y_{k+1} \quad \forall k \in \{1, \ldots, K-1\}, \quad \sum_{k=1}^{K} y_k = 1, \quad y \in \{0,1\}^K. \tag{18b}$$

We can also adapt Inc, Log and SOS2. For Log we replace (18b) by the corresponding constraints (9c), which in this case are $\sum_{k\in L_s} \lambda_k \leq y_s$, $\sum_{k\in R_s} \lambda_k \leq (1-y_s)$ and $y_s \in \{0,1\}$ for all $s \in \{1, \ldots, \lceil \log_2(K) \rceil\}$, where $L_s := \left\{k \in \{0, \ldots, K\} : (k=0 \text{ or } G_l^k = 1) \text{ and } \left(k = K \text{ or } G_l^{k+1} = 1\right)\right\}$ and $R_s := \left\{k \in \{0, \ldots, K\} : (k=0 \text{ or } G_l^k = 0) \text{ and } \left(k = K \text{ or } G_l^{k+1} = 0\right)\right\}$ for an arbitrary but fixed set of vectors $(G^l)_{l=1}^K \subset \{0,1\}^{\lceil \log_2(K) \rceil}$ that form a Gray code. For Inc we obtain the formulation given by

$$d_0 + \sum_{k=1}^{K} \delta_k \left(d_k - d_{k-1}\right) = x, \quad f_0 + \sum_{k=1}^{K} \delta_k \left(f_k - f_{k-1}\right) \leq z, \tag{19a}$$

$$\delta_1 \leq 1, \quad \delta_K \geq 0, \quad \delta_{k+1} \leq y_k \leq \delta_k, \quad y_k \in \{0,1\} \quad \forall k \in \{1, \ldots, K-1\}. \tag{19b}$$

For SOS2 the adaptation is analoguous to the one for CC and is described in Vielma et al. (2008). We denote these models CC Dup, Inc Dup, Log Dup and SOS2 Dup. In Appendix EC.4 we illustrate them for the function depicted in Figure 3(a).

The second technique can be applied when all discontinuities of f are caused by fixed charge type jumps. In this case, f is the sum of a continuous function f_C of the form (2) and a lower semicontinuous non-decreasing step function

$$f_J(x) := \begin{cases} 0 & x = 0 \\ b_k & x \in (d_{k-1}, d_k] \quad \forall k \in \{1, \ldots, K\} \end{cases} \tag{20}$$

for $(d_k)_{k=0}^K \in \mathbb{R}^{K+1}$, $(b_k)_{k=1}^K \in \mathbb{R}_+^K$ such that $0 = d_0 < d_1 < \ldots < d_K = u$ and $0 \leq b_1 \leq b_2 \leq \ldots \leq b_K$. Hence, for $(m_k)_{k=1}^K \in \mathbb{R}^K$ and $(c_k)_{k=1}^K \in \mathbb{R}^K$, f can be described as

$$f(x) := \begin{cases} c_1 & x = 0 \\ m_k x + c_k + b_k & x \in (d_{k-1}, d_k] \quad \forall k \in \{1, \ldots, K\}. \end{cases} \tag{21}$$

By using the relation $f = f_C + f_J$ we can construct a model for epi(f) from models for epi(f_C) and epi(f_J). This combination of models is referred to as *model linkage* in Jeroslow and Lowe (1985) where it is shown to computationally perform relatively poorly, in part because formulation sharpness is not preserved by model linkage and in part because of poor coordination between the binary variables of the linked models. Fortunately, as noted in Lowe (1984), it is sometimes possible to improve model coordination by using ad-hoc techniques. We illustrate this possible coordination by using two specific examples. In both cases we need a lower semicontinuous function $f : [0, u] \to \mathbb{R}$ which is continuous and zero valued at zero and hence has $0 = c_1 = b_1$ in characterizations (20) and (21). The first coordination is for the model obtained by linking CC and the model of f_J given by

$$\sum_{k=0}^{K} d_k \lambda_{d_k} = x, \quad \sum_{k=1}^{K} b_k w_k \leq z, \quad \sum_{k=0}^{K} \lambda_{d_k} = 1, \quad \sum_{k=1}^{K} w_k = 1, \quad 0 \leq \lambda_{d_0} \leq w_1, \quad 0 \leq \lambda_{d_K} \leq w_K \tag{22a}$$

$$0 \leq \lambda_{d_k} \leq (w_k + w_{k+1}) \ \forall k \in \{1, \ldots, K-1\}, \quad w_k \in \{0,1\} \ \forall k \in \{1, \ldots, K\}. \tag{22b}$$

To coordinate we identify the λ_{d_k} variables of the models and force $w_k = y_{[d_{k-1}, d_k]}$. The resulting model is given by

$$\sum_{k=0}^{K} d_k \lambda_{d_k} = x, \ \lambda_{d_0} m_1 d_0 + \sum_{k=1}^{K} \Big(\lambda_{d_k} (m_k d_k + c_k) + b_k w_k \Big) \leq z, \ \sum_{k=0}^{K} \lambda_{d_k} = 1, \ 0 \leq \lambda_{d_0} \leq w_1 \tag{23a}$$

$$0 \leq \lambda_{d_K} \leq w_K, \quad 0 \leq \lambda_{d_k} \leq (w_k + w_{k+1}) \quad \forall k \in \{1, \ldots, K-1\}, \quad \sum_{k=1}^{K} w_k = 1, \quad w \in \{0,1\}^K. \tag{23b}$$

We refer to this formulation as the coordinated convex combination model and denote it by CC Coord. A similar coordination can be achieved by linking Inc and another model of f_J. The resulting model is given by

$$\sum_{k=1}^{K} \delta_k (d_k - d_{k-1}) = x, \qquad \sum_{k=1}^{K} \big(m_k d_k - m_k d_{k-1} \big) \delta_k + \sum_{k=1}^{K-1} \big(b_{k+1} - b_k \big) y_k \leq z \tag{24a}$$

$$\delta_1 \leq 1, \quad \delta_K \geq 0, \quad \delta_{k+1} \leq y_k \leq \delta_k, \quad y_k \in \{0,1\} \quad \forall k \in \{1, \ldots, K-1\}. \tag{24b}$$

This model has been studied in Keha (2003). We refer to this formulation as the coordinated incremental model and denote it by Inc Coord. We illustrate these formulations in Appendix EC.4.

Regarding the properties of the formulations, it is direct that Proposition 1, Theorem 2 and Theorem 4 also hold for lower semicontinuous piecewiselinear functions. It is also direct that DCC, DLog and MC remain locally ideal for lower semicontinuous functions, that Inc Dup, Log Dup and Inc Coord are locally ideal and that CC Dup is sharp, but not locally ideal. Finally, it is direct that CC Coord is not locally ideal, but the following proposition holds.

PROPOSITION 2. *CC Coord is sharp.*

7. Computational Experiments for Lower Semicontinuous Functions

In this section we computationally test the MIP formulations for lower semicontinuous piecewise linear functions. We use the same transportation problems from Section 5.

7.1. Discontinuous Separable Functions

The first set of experiments considers formulations for univariate lower semicontinuous functions. The instances tested in this section were obtained from the transportation problems from Section 5.1 by modifying functions $f_e(x_e)$ of the flow x_e on arc e. Each function $f_e(x_e)$ affine in segments $\{[d_{k-1}, d_k]\}_{k=1}^K$ was transformed into a discontinuous function by adding fixed charge jumps in each of the breakpoints $\{d_k\}_{k=0}^K$. Each jump was randomly generated by independently selecting an integer in $[10, 50]$ using a uniform distribution.

We tested MC, DCC and DLog as they can directly handle lower semicontinuous functions. However, we modified DLog as it initially performed poorly (for $K = 4$ it had an average solve time of 562 seconds and a maximum solve time of 6615 seconds). We believe that this poor performance was due to $|\mathcal{P}|$ not being a power of two (for $K = 4$ we have $\mathcal{P} = \{d_0 = 0, (d_0, d_1], (d_1, d_2], (d_2, d_3], (d_3, d_4]\}$) as this is a common problem with binary encoded formulations (Coppersmith and Lee 2005). To resolve this we subtracted $f_e^+(0)$ from each function $f_e(x_e)$ and reset the value of $f_e(0)$ to 0. This eliminated the fixed charges at 0 leaving each $f_e(x_e)$ continuous and zero valued at 0. To restore the fixed charges we added a binary variable $y^e \in \{0, 1\}$ for each $e \in E$ with objective coefficient equal to the original fixed charge $f_e^+(0)$ and constraint $x_e \leq u_e y^e$. We also tested CC Coord and Inc Coord with the fixed charge elimination technique because they require functions that are continuous and zero valued at 0. We additionally tested the formulations obtained by applying the break point duplication technique to CC, Log, Inc and SOS2. Additional combinations of models and techniques are not included either because they are redundant (e.g. DCC directly handles lower semicontinuous functions and hence does not require the break point duplication technique) or because they are not compatible (e.g. we are not aware of any effective coordination technique for Log). Table 5 shows the usual statistics for these instances.

model	min	**avg**	max	std	win	fail
MC	0.5	**5.5**	30	5.2	76	0
Inc Coord	0.8	**7.3**	40	6.3	15	0
DLog FC	0.8	**9.0**	41	6.5	6	0
Inc Dup	1.0	**10.7**	61	8.4	3	0
Log Dup	1.0	**13.0**	69	8.7	0	0
DCC	2.0	**14.8**	75	9.5	0	0
CC Coord	1.1	**15.7**	116	14.1	0	0
SOS2 Dup	3.2	**56.7**	522	75.3	0	0
CC Dup	7.4	**78.9**	646	105.3	0	0

(a) 4 segments.

model	min	**avg**	max	std	win	fail
MC	0.0	**16**	107	23	86	0
DLog FC	0.3	**32**	123	21	9	0
Log Dup	2.1	**43**	241	38	4	0
Inc Coord	7.9	**70**	298	51	0	0
Inc Dup	18.7	**84**	300	51	0	0
DCC	0.0	**366**	10000	1110	1	1
SOS2 Dup	8.8	**476**	5919	853	0	0
CC Coord	21.3	**699**	5438	1014	0	0
CC Dup	8.1	**895**	10000	1644	0	2

(b) 8 segments.

model	min	**avg**	max	std	win	fail
DLog FC	23	**106**	445	88	55	0
MC	13	**263**	2697	401	29	0
Log Dup	12	**331**	10000	1055	16	1
Inc Coord	108	**333**	2037	247	0	0
Inc Dup	105	**405**	1548	278	0	0
SOS2 Dup	51	**1952**	10000	2587	0	6
CC Dup	177	**4409**	10000	3223	0	18
CC Coord	342	**6018**	10000	3624	0	36
DCC	110	**8046**	10000	3551	0	76

(c) 16 segments.

model	min	**avg**	max	std	win	fail
DLog FC	54	**779**	5395	958	84	0
Inc Coord	287	**1586**	10000	1457	1	1
Inc Dup	315	**1935**	10000	1984	2	4
Log Dup	77	**2661**	10000	3268	4	12
MC	116	**4282**	10000	4070	9	30

(d) 32 segments.

Table 5 Solve times for univariate discontinuous functions [s].

Again MC is one of the best performers except for $K = 32$ where the logarithmic models again have the advantage. The duplication and coordination techniques only seem to work well for Inc and Log which were already faster than CC in the continuous case. This could explain their advantage when using the duplication and coordination techniques as well. However, this explanation does not hold for SOS2, which did very well in the continuous case, but performed poorly here.

7.2. Discontinuous Non-Separable Functions

The set of experiments in this section considers non-separable functions of two variables. The instances tested in this section where obtained from the 5×2 multi commodity transportation problems from Section 5.2 by replacing function $f_e(x_e^1, x_e^2)$ of the flows x_e^i in arc e of commodity i

for $i=1,2$. To define the new function we use the $K\times K$ grid $\{d_0^1,\ldots,d_K^1\}\times\{d_0^2,\ldots,d_K^2\}$ obtained from the subdivision of $[0,u_e^1]$ and $[0,u_e^2]$ into K intervals as determined from the respective original transportation problems. We select two random samples of size K from set $\{0,1,\ldots,10K-1\}$ and sort them in non-increasing order to obtain $(r_k^i)_{k=1}^K$ for each $i=1,2$. We then define $s_0^i=0$ and $s_k^i=r_k^i(d_k^i-d_{k-1}^i)+s_{k-1}^i$ for each $k\in\{1,\ldots,K\}$ and $i=1,2$. $f_e(x_e^1,x_e^2)$ is defined as

$$f_e(x_e^1,x_e^2):=\begin{cases} x_e^1+x_e^2 & (x,y)=(0,0)\\ x_e^1+x_e^2+s_k^1 & x\in(d_{k-1}^1,d_k^1],\quad y=0\\ x_e^1+x_e^2+s_k^2 & y\in(d_{k-1}^2,d_k^2],\quad x=0\\ x_e^1+x_e^2+0.75(s_k^1+s_l^2) & (x,y)\in(d_{k-1}^1,d_k^1]\times(d_{l-1}^2,d_l^2]. \end{cases}$$

The idea is that for each commodity there is a fixed shipping charge for arc e that depends on the interval $(d_{k-1}^i,d_k^i]$ in which the amount x_e^i shipped falls. We have that this fixed charge divided by the amount shipped is non-increasing because of economies of scale and that if both commodities are shipped through arc e there is a 75% discount on the sum of the fixed charges.

We only tested MC, DCC and DLog as they can handle general lower semicontinuous piecewise linear functions. Table 6 shows the usual statistics for different grid sizes. We again see that MC is always faster than DCC and is only significantly slower than DLog for the largest grids. Finally, we note that the smaller solve times for these instances when compared to the ones in Section 5.2 could be due to the fact that here the only nonlinearities in the objective functions are fixed charges.

stat	min	**avg**	max	std	wins	fail
MC	0.1	**2.3**	8.8	1.8	97	0
DLog	0.4	**6.0**	19.3	3.9	3	0
DCC	0.9	**9.9**	29.8	6.6	0	0

(a) 4×4 grid.

stat	min	**avg**	max	std	wins	fail
DLog	1.1	**17**	59	11	51	0
MC	1.0	**19**	122	18	49	0
DCC	8.4	**83**	377	64	0	0

(b) 8×8 grid.

stat	min	**avg**	max	std	wins	fail
DLog	4.8	**55**	201	36	96	0
MC	10.2	**209**	1138	195	4	0
DCC	51.2	**890**	2993	542	0	0

(c) 16×16 grid.

stat	min	**avg**	max	std	wins	fail
DLog	56	**319**	1385	201	100	0
MC	151	**4310**	10000	3780	0	25
DCC	1648	**8504**	10000	2545	0	65

(d) 32×32 grid.

Table 6 Solve times for non-separable functions [s].

8. Conclusions

We studied the modeling of piecewise linear functions as MIPs. We reviewed several new and existing formulations for continuous functions with particular attention paid to their extension to the multivariate non-separable case. We also compared these formulations both with respect to their theoretical properties and their relative computational performance. In addition we studied several ways to extend these formulations to consider lower semicontinuous functions.

Because of the limited computational experiments it is hard to reach categorical conclusions. However there are several trends that, combined with the theoretical properties of the formulations, provide general guidelines for the use of the different formulations by practitioners. For example, when the number of polytopes defining the piecewise linear function is small MC seems to be one of the best choices. Furthermore it seems to be always preferable to CC and DCC. Another example concerns functions defined by a large number of polytopes. In this case the sizes of logarithmic formulations DLog and Log can give them a significant computational advantage. Finally, for lower semicontinuous functions it seems that, with the exception of SOS2 Dup, special ad-hoc techniques only provide an advantage when they are used to adapt formulations that already performed well in the continuous case.

Acknowledgments

This research has been supported by NSF grants CMMI-0522485 and CMMI-0758234, AFOSR grant FA9550-07-1-0177 and Exxon Mobil Upstream Research Company. The authors would also like to thank an anonymous associate editor and two anonymous referees for their prompt review and thoughtful comments.

References

Aichholzer, O., F. Aurenhammer, F. Hurtado, H. Krasser. 2003. Towards compatible triangulations. *Theoret. Comput. Sci.* **296** 3–13.

Balakrishnan, A., S. C. Graves. 1989. A composite algorithm for a concave-cost network flow problem. *Networks* **19** 175–202.

Balas, E. 1979. Disjunctive programming. *Ann. of Discrete Math.* **5** 3–51.

Bergamini, M. L., P. Aguirre, I. Grossmann. 2005. Logic-based outer approximation for globally optimal synthesis of process networks. *Comput. and Chemical Engrg.* **29** 1914–1933.

Bergamini, M. L., I. Grossmann, N. Scenna, P. Aguirre. 2008. An improved piecewise outer-approximation algorithm for the global optimization of minlp models involving concave and bilinear terms. *Comput. and Chemical Engrg.* **32** 477–493.

Carnicer, J. M., M. S. Floater. 1996. Piecewise linear interpolants to Lagrange and Hermite convex scattered data. *Numer. Algorithms* **13** 345–364.

Coppersmith, D., J. Lee. 2005. Parsimonious binary-encoding in integer programming. *Discrete Optim.* **2** 190–200.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2003a. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Sci.* **49** 1268–1273.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2003b. Models and methods for merge-in-transit operations. *Transportation Sci.* **37** 1–22.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2007. Variable disaggregation in network flow problems with piecewise linear costs. *Oper. Res.* **55** 146–157.

Dantzig, G. B. 1960. On the significance of solving linear-programming problems with some integer variables. *Econometrica* **28** 30–44.

Dantzig, G. B. 1963. *Linear Programming and Extensions*. Princeton University Press, Princeton.

de Farias Jr., I. R., M. Zhao, H. Zhao. 2008. A special ordered set approach for optimizing a discontinuous separable piecewise linear function. *Oper. Res. Lett.* **36** 234–238.

Fourer, R., D. M. Gay, B. W. Kernighan. 1993. *AMPL–A Modeling Language for Mathematical Programming*. The Scientific Press.

Garfinkel, R. S., G. L. Nemhauser. 1972. *Integer Programming*. Wiley.

Graf, T., P. Vanhentenryck, C. Pradelleslasserre, L. Zimmer. 1990. Simulation of hybrid circuits in constraint logic programming. *Comput. & Math. With Appl.* **20** 45–56.

Ibaraki, T. 1976. Integer programming formulation of combinatorial optimization problems. *Discrete Math.* **16** 39–52.

Jeroslow, R. G. 1987. Representability in mixed integer programming 1: characterization results. *Discrete Appl, Math.* **17** 223–243.

Jeroslow, R. G. 1989. Representability of functions. *Discrete Appl, Math.* **23** 125–137.

Jeroslow, R. G., J. K. Lowe. 1984. Modeling with integer variables. *Math. Programming Stud.* **22** 167–184.

Jeroslow, R. G., J. K. Lowe. 1985. Experimental results on the new techniques for integer programming formulations. *J. of the Oper. Res. Soc.* **36** 393–403.

Kannan, R. 1992. Lattice translates of a polytope and the frobenius problem. *Combinatorica* **12** 161–177.

Keha, A. B. 2003. A polyhedral study of nonconvex piecewise linear optimization. Ph.D. thesis, Georgia Institute of Technology.

Keha, A. B., I. R. de Farias, G. L. Nemhauser. 2004. Models for representing piecewise linear cost functions. *Oper. Res. Lett.* **32** 44–48.

Keha, A. B., I. R. de Farias, G. L. Nemhauser. 2006. A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Oper. Res.* **54** 847–858.

Lasdon, L. S., A. D. Waren. 1980. A survey of nonlinear programming applications. *Oper. Res.* **28** 1029–1073.

Lee, J., D. Wilson. 2001. Polyhedral methods for piecewise-linear functions I: the lambda method. *Discrete Appl, Math.* **108** 269–285.

Lowe, J. K. 1984. Modelling with integer variables. Ph.D. thesis, Georgia Institute of Technology.

Magnanti, T. L., D. Stratila. 2004. Separable concave optimization approximately equals piecewise linear optimization. G. L. Nemhauser, D. Bienstock, eds., *IPCO*, *Lecture Notes in Computer Science*, vol. 3064. Springer, 234–243.

Markowitz, H., A. Manne. 1957. On the solution of discrete programming-problems. *Econometrica* **25** 84–110.

Martin, A., M. Moller, S. Moritz. 2006. Mixed integer models for the stationary case of gas network optimization. *Math. Programming* **105** 563–582.

Meyer, R. R. 1976. Mixed integer minimization models for piecewise-linear functions of a single variable. *Discrete Math.* **16** 163–171.

Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and combinatorial optimization*. Wiley-Interscience.

Padberg, M. 2000. Approximating separable nonlinear functions via mixed zero-one programs. *Oper. Res. Lett.* **27** 1–5.

Padberg, M. W., M. P. Rijal. 1996. *Location, Scheduling, Design, and Integer Programming*. Springer.

Pottmann, H., R. Krasauskas, B. Hamann, K. I. Joy, W. Seibold. 2000. On piecewise linear approximation of quadratic functions. *Journal for Geometry and Graphics* **4** 9–31.

Sherali, H. D. 2001. On mixed-integer zero-one representations for separable lower-semicontinuous piecewise-linear functions. *Oper. Res. Lett.* **28** 155–160.

Todd, M. J. 1977. Union Jack triangulations. S. Karamardian, ed., *Fixed Points: algorithms and applications*. Academic Press, 315–336.

Tomlin, J.A. 1981. A suggested extension of special ordered sets to non-separable non-convex programming problems. P. Hansen, ed., *Studies on Graphs and Discrete Programming*, *Annals of Discrete Mathematics*, vol. 11. North Holland, 359–370.

Vajda, S. 1964. *Mathematical Programming*. Addison-Wesley.

Vielma, J. P., A. B. Keha, G. L. Nemhauser. 2008. Nonconvex, lower semicontinuous piecewise linear optimization. *Discrete Optim.* **5** 467–488.

Vielma, J. P., G. L. Nemhauser. 2008a. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. A. Lodi, A. Panconesi, G. Rinaldi, eds., *IPCO*, *Lecture Notes in Computer Science*, vol. 5035. Springer, 199–213.

Vielma, J. P., G. L. Nemhauser. 2008b. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Georgia Institute of Technology* .

Wilf., H. S. 1989. *Combinatorial algorithms–an update*, *CBMS-NSF regional conference series in applied mathematics*, vol. 55. Society for Industrial and Applied Mathematics.

Wilson, D. L. 1998. Polyhedral methods for piecewise-linear functions. Ph.D. thesis, University of Kentucky.

Appendices

EC.1. Example for Continuous Functions

The function given in Figure 1(a) can be described as

$$f(x) := \begin{cases} 22x + 10 & x \in [0,1] \\ 8x + 24 & x \in [1,2] \\ -17.5x + 75 & x \in [2,4] \\ 10x - 35 & x \in [4,5] \end{cases} \tag{EC.1}$$

and characterization (4) of its epigraph is given by

$$\begin{aligned} \operatorname{epi}(f) = \{(0,r) : r \geq 0\} + \Big(&\operatorname{conv}\big(\{(0,10),(1,32)\}\big) \cup \operatorname{conv}\big(\{(1,32),(2,40)\}\big) \\ &\cup \operatorname{conv}\big(\{(2,40),(4,5)\}\big) \cup \operatorname{conv}\big(\{(4,5),(5,15)\}\big)\Big). \end{aligned}$$

Note that this representation can be simplified by replacing $\operatorname{conv}(\{(2,40),(4,5)\}) \cup \operatorname{conv}(\{(4,5),(5,15)\})$ with $\operatorname{conv}(\{(2,40),(4,5),(5,15)\})$, but this requires detecting that $(\operatorname{conv}(\{(2,40),(4,5)\}) + \{(0,r) : r \geq 0\}) \cup (\operatorname{conv}(\{(4,5),(5,15)\}) + \{(0,r) : r \geq 0\})$ is in fact a polyhedron.

We now describe the formulations in Section 3 for the function defined in (EC.1). DCC is given by

$$0\lambda_{[0,1],0} + 1\left(\lambda_{[0,1],1} + \lambda_{[1,2],1}\right) + 2\left(\lambda_{[1,2],2} + \lambda_{[2,4],2}\right) + 4\left(\lambda_{[2,4],4} + \lambda_{[4,5],4}\right) + 5\lambda_{[4,5],5} = x$$

$$10\lambda_{[0,1],0} + 32\left(\lambda_{[0,1],1} + \lambda_{[1,2],1}\right) + 40\left(\lambda_{[1,2],2} + \lambda_{[2,4],2}\right) + 5\left(\lambda_{[2,4],4} + \lambda_{[4,5],4}\right) + 15\lambda_{[4,5],5} \leq z$$

$$\lambda_{[0,1],0},\ \lambda_{[0,1],1},\ \lambda_{[1,2],1},\ \lambda_{[1,2],2},\ \lambda_{[2,4],2},\ \lambda_{[2,4],4},\ \lambda_{[4,5],4},\ \lambda_{[4,5],5} \geq 0$$

$$\lambda_{[0,1],0} + \lambda_{[0,1],1} = y_{[0,1]}, \quad \lambda_{[1,2],1} + \lambda_{[1,2],2} = y_{[1,2]}, \quad \lambda_{[2,4],2} + \lambda_{[2,4],4} = y_{[2,4]}, \quad \lambda_{[4,5],4} + \lambda_{[4,5],5} = y_{[4,5]}$$

$$y_{[0,1]} + y_{[1,2]} + y_{[2,4]} + y_{[4,5]} = 1, \quad y_{[0,1]},\ y_{[1,2]},\ y_{[2,4]},\ y_{[4,5]} \in \{0,1\}.$$

For $B([0,1]) = (0,0)^T$, $B([1,2]) = (0,1)^T$, $B([2,4]) = (1,1)^T$, $B([4,5]) = (1,0)^T$ DLog is given by

$$0\lambda_{[0,1],0} + 1\left(\lambda_{[0,1],1} + \lambda_{[1,2],1}\right) + 2\left(\lambda_{[1,2],2} + \lambda_{[2,4],2}\right) + 4\left(\lambda_{[2,4],4} + \lambda_{[4,5],4}\right) + 5\lambda_{[4,5],5} = x$$

$$10\lambda_{[0,1],0}+32\left(\lambda_{[0,1],1}+\lambda_{[1,2],1}\right)+40\left(\lambda_{[1,2],2}+\lambda_{[2,4],2}\right)+5\left(\lambda_{[2,4],4}+\lambda_{[4,5],4}\right)+15\lambda_{[4,5],5}\leq z$$

$$\lambda_{[0,1],0},\,\lambda_{[0,1],1},\,\lambda_{[1,2],1},\,\lambda_{[1,2],2},\,\lambda_{[2,4],2},\,\lambda_{[2,4],4},\,\lambda_{[4,5],4},\,\lambda_{[4,5],5}\geq 0$$

$$\lambda_{[2,4],2}+\lambda_{[2,4],4}+\lambda_{[4,5],4}+\lambda_{[4,5],5}\leq y_1,\quad \lambda_{[0,1],0}+\lambda_{[0,1],1}+\lambda_{[1,2],1}+\lambda_{[1,2],2}\leq (1-y_1)$$

$$\lambda_{[1,2],1}+\lambda_{[1,2],2}+\lambda_{[2,4],2}+\lambda_{[2,4],4}\leq y_2,\quad \lambda_{[0,1],0}+\lambda_{[0,1],1}+\lambda_{[4,5],4}+\lambda_{[4,5],5}\leq (1-y_2)$$

$$y_1,\,y_2\in\{0,1\}.$$

CC is given by

$$0\lambda_0+1\lambda_1+2\lambda_2+4\lambda_4+5\lambda_5=x,\quad 10\lambda_0+32\lambda_1+40\lambda_2+5\lambda_4+15\lambda_5\leq z$$

$$\lambda_0,\,\lambda_1,\,\lambda_2,\,\lambda_4,\,\lambda_5\geq 0,\quad \lambda_0+\lambda_1+\lambda_2+\lambda_4+\lambda_5=1$$

$$\lambda_0\leq y_{[0,1]},\quad \lambda_1\leq y_{[0,1]}+y_{[1,2]},\quad \lambda_2\leq y_{[1,2]}+y_{[2,4]},\quad \lambda_4\leq y_{[2,4]}+y_{[4,5]},\quad \lambda_5\leq y_{[4,5]}$$

$$y_{[0,1]}+y_{[1,2]}+y_{[2,4]}+y_{[4,5]}=1,\quad y_{[0,1]},\,y_{[1,2]},\,y_{[2,4]},\,y_{[4,5]}\in\{0,1\}.$$

For $G^1=(0,0)^T$, $G^2=(1,0)^T$, $G^3=(1,1)^T$, $G^4=(0,1)^T$ and $\varphi(0)=0$, $\varphi(1)=1$, $\varphi(2)=2$, $\varphi(3)=4$, $\varphi(4)=5$, Log is given by

$$0\lambda_0+1\lambda_1+2\lambda_2+4\lambda_3+5\lambda_4=x,\quad 10\lambda_0+32\lambda_1+40\lambda_2+5\lambda_3+15\lambda_4\leq z$$

$$\lambda_0,\,\lambda_1,\,\lambda_2,\,\lambda_3,\,\lambda_4\geq 0,\quad \lambda_0+\lambda_1+\lambda_2+\lambda_3+\lambda_4=1$$

$$\lambda_2\leq y_1,\quad \lambda_0+\lambda_4\leq(1-y_1),\quad \lambda_3+\lambda_4\leq y_2,\quad \lambda_0+\lambda_1\leq(1-y_2),\quad y_1,\,y_2\in\{0,1\}.$$

Finally, MC is given by

$$x^{[0,1]}+x^{[1,2]}+x^{[2,4]}+x^{[4,5]}=x$$

$$\left(22x^{[0,1]}+10y_{[0,1]}\right)+\left(8x^{[1,2]}+24y_{[1,2]}\right)+\left(-17.5x^{[2,4]}+75y_{[2,4]}\right)+\left(10x^{[4,5]}-35y_{[4,5]}\right)\leq z$$

$$0y_{[0,1]}\leq x^{[0,1]}\leq y_{[0,1]},\quad 1y_{[1,2]}\leq x^{[1,2]}\leq 2y_{[1,2]},\quad 2y_{[2,4]}\leq x^{[2,4]}\leq 4y_{[2,4]},\quad 4y_{[4,5]}\leq x^{[4,5]}\leq 5y_{[4,5]}$$

$$y_{[0,1]}+y_{[1,2]}+y_{[2,4]}+y_{[4,5]}=1,\quad y_{[0,1]},\,y_{[1,2]},\,y_{[2,4]},\,y_{[4,5]}\in\{0,1\}$$

and Inc is given by

$$10+22\delta_1+8\delta_2-35\delta_3+10\delta_4\leq z,\quad 0+\delta_1+\delta_2+2\delta_3+\delta_4=x$$

$$y_1\leq\delta_1\leq 1,\quad y_2\leq\delta_2\leq y_1,\quad y_3\leq\delta_3\leq y_2,\quad 0\leq\delta_4\leq y_3,\quad y_1,\,y_2,\,y_3\in\{0,1\}.$$

EC.2. Proofs

THEOREM 1. *All formulations from Section 3 except CC are locally ideal.*

Proof of Theorem 1. All models except CC, DLog and Log have been previously shown to be locally ideal (Balas 1979, Jeroslow and Lowe 1984, Lowe 1984, Padberg 2000, Sherali 2001, Wilson 1998), so we only need to prove that DLog and Log are locally ideal.

For Log assume for contradiction that there exists an vertex (x,z,λ,y) of (9) such that $y_s \in (0,1)$ for some $s \in S$. We divide the proof in two main cases.

Case 1: $\sum_{v \in L_s} \lambda_v < y_s$ **and** $\sum_{v \in R_s} \lambda_v < (1-y_s)$**.** For $\varepsilon > 0$ define (x^1,z^1,λ^1,y^1) and (x^2,z^2,λ^2,y^2) as $x^1 = x^2 = x,\ z^1 = z^2 = z,\ \lambda^1 = \lambda^2 = \lambda,\ y^1 = y + \varepsilon$ and $y^2 = y - \varepsilon$. For sufficiently small ε we have that (x^1,z^1,λ^1,y^1) and (x^2,z^2,λ^2,y^2) comply with (9) and $(x,z,\lambda,y) = 1/2(x^1,z^1,\lambda^1,y^1) + 1/2(x^2,z^2,\lambda^2,y^2)$. This contradicts (x,z,λ,y) being a vertex.

Case 2: $\sum_{v \in L_s} \lambda_v = y_s$ **or** $\sum_{v \in R_s} \lambda_v = (1-y_s)$**.** Without loss of generality we may assume that $\sum_{v \in L_s} \lambda_v = y_s$. We then have $v_s \in L_s$ such that $0 < \lambda_{v_s} < 1$ and $v_l \notin L_s$ such that $0 < \lambda_{v_l} < 1$. If $\sum_{v \in R_s} \lambda_v = (1-y_s)$ we additionally select $v_l \in R_s$. For $\varepsilon > 0$ we define (x^1,z^1,λ^1,y^1) and (x^2,z^2,λ^2,y^2) in the following way. First let $\lambda^1_k = \lambda^1_k = \lambda_k$ for all $k \notin \{v_s, v_l\}$, $\lambda^1_{v_s} = \lambda_{v_s} + \varepsilon$, $y^1_s = y_s + \varepsilon$, $\lambda^2_{v_s} = \lambda_{v_s} - \varepsilon$, $y^2_s = y_s - \varepsilon$, $\lambda^1_{v_l} = \lambda^1_{v_l} - \varepsilon$ and $\lambda^2_{v_l} = \lambda^2_{v_l} + \varepsilon$. To define y^1_t and y^2_t for each $t \in S \setminus \{s\}$ we only need to consider the following four cases (note that $L_t \cap R_t = \emptyset$ and that without loss of generality we can exchange R_t and L_t):

(a) $v_s, v_l \in L_t$ and $v_s, v_l \notin R_t$.

(b) $v_s \in L_t$ and $v_l \in R_t$.

(c) $v_s \in L_t$, $v_l \notin L_t$ and $v_l \notin R_t$ (case $v_l \in L_t$, $v_s \notin L_t$ and $v_s \notin R_t$ is analogous).

(d) v_s, $v_l \notin L_t$ and $v_s, v_l \notin R_t$.

For case a) we can simply set $y^1_t = y^2_t = y$. For case b) we have $0 < y_t < 1$ and we can set $y^1_t = y_t + \varepsilon$ and $y^2_t = y_t - \varepsilon$. For case c) we either have $\sum_{v \in L_t} \lambda_v < y_t$ or $\sum_{v \in L_t} \lambda_v = y_t$. For the first case we can simply set $y^1_t = y^2_t = y$. For the second case we have $0 < y_t < 1$ and $\sum_{v \in R_t} \lambda_v < (1-y_t)$ and we can set $y^1_t = y_t + \varepsilon$ and $y^2_t = y_t - \varepsilon$. For case d) we can set $y^1_t = y^2_t = y$. Finally we set

$x^1 = x + \varepsilon(v_s - v_l)$, $x^2 = x - \varepsilon(v_s - v_l)$, $z^1 = z + \varepsilon(f(v_s) - f(v_l))$ and $z^2 = z - \varepsilon(f(v_s) - f(v_l))$. We again have that for sufficiently small ε $(x^1, z^1, \lambda^1, y^1)$ and $(x^2, z^2, \lambda^2, y^2)$ comply with (9) and $(x, z, \lambda, y) = 1/2(x^1, z^1, \lambda^1, y^1) + 1/2(x^2, z^2, \lambda^2, y^2)$.

For DLog the proof is analogous. □

PROPOSITION 1. *Any locally ideal formulation is sharp.*

Proof of Proposition 1. We need to prove $P_{(x,y)} \subset \operatorname{conv}(\operatorname{epi}(f))$. If $x \in P_{(x,y)}$ then because P is locally ideal there exist $\lambda \in \mathbb{R}^p$, $y \in [0,1]^q$ such that $(x, z, \lambda, y) = (0, h, 0, 0) + \sum_{i \in I} \mu_i (x^i, z^i, \lambda^i, y^i)$ for $h \geq 0$, $|I| < \infty$, $\mu \in \mathbb{R}^I_+$, $\sum_{i \in I} \mu_i = 1$, and $(x^i, z^i, \lambda^i, y^i) \in P$ with $y^i \in \{0,1\}^q$ for every $i \in I$. Then by (1) $(x^i, z^i) \in \operatorname{epi}(f)$ for all $i \in I$ and hence $(x, z) \in \operatorname{conv}(\operatorname{epi}(f))$. □

THEOREM 3. *All formulations from Section 3 are sharp.*

Proof of Theorem 3. This is direct from Theorem 1 for all formulations except CC. For CC the result follows by noting that the projection onto the x and z variables of the polyhedron given by $\sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v v = x$, $\sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v f(v) \leq z$, $\lambda_v \geq 0 \quad \forall v \in \mathcal{V}(\mathcal{P})$ and $\sum_{v \in \mathcal{V}(\mathcal{P})} \lambda_v = 1$ is clearly contained in $\operatorname{conv}(\operatorname{epi}(f))$. □

PROPOSITION 2. *CC Coord is sharp.*

Proof of Proposition 2. It suffices to show that for f defined in (21) and for any vertex (λ^*, w^*) of

$$\sum_{k=0}^{K} \lambda_{d_k} = 1, \quad \sum_{k=1}^{K} w_k = 1, \quad w \in \{0,1\}^K \tag{EC.2a}$$

$$0 \leq \lambda_{d_0} \leq w_1, \quad 0 \leq \lambda_{d_K} \leq w_K, \quad 0 \leq \lambda_{d_k} \leq (w_k + w_{k+1}) \quad \forall k \in \{1, \ldots, K-1\} \tag{EC.2b}$$

we have $(x^*, \underline{z}^*) \in \operatorname{conv}(\operatorname{epi}(f))$ for $\underline{z}^* := \lambda^*_{d_0} m_1 d_0 + \sum_{k=1}^{K} \left(\lambda^*_{d_k} (m_k d_k + c_k) + b_k w^*_k\right)$, $x^* := \sum_{k=0}^{K} d_k \lambda^*_{d_k}$.

From Proposition 4 of Lee and Wilson (2001) we have that the vertices of (EC.2) are of the following forms:

1. $\lambda^*_{d_l} = w^*_l = 1, \quad \lambda^*_{d_k} = 0 \quad \forall k \neq l, \quad w^*_k = 0 \quad \forall k \neq l.$

2. $\lambda^*_{d_l} = w^*_{l+1} = 1, \quad \lambda^*_{d_k} = 0 \quad \forall k \neq l, \quad w^*_k = 0 \quad \forall k \neq l+1.$

3. $\lambda^*_{d_{l-1}} = \lambda^*_{d_l} = w^*_l = w^*_{l+1} = 1/2, \quad \lambda^*_{d_k} = 0 \quad \forall k \notin \{l-1, l\}, \quad w^*_k = 0 \quad \forall k \notin \{l, l+1\}.$

4. $\lambda^*_{d_{l-1}} = \lambda^*_{d_l} = w^*_{l-1} = w^*_l = 1/2, \quad \lambda^*_{d_k} = 0 \quad \forall k \notin \{l-1, l\}, \quad w^*_k = 0 \quad \forall k \notin \{l-1, l\}.$

For case 1 we have $x^* = d_l$ and $\underline{z}^* = m_l d_l + c_l + b_l$ so $(x^*, \underline{z}^*) \in \text{epi}(f)$. For case 2 and $l \geq 1$ we have $x^* = d_l$ and $\underline{z}^* = m_l d_l + c_l + b_{l+1} \geq m_l d_l + c_l + b_l$ so $(x^*, \underline{z}^*) \in \text{epi}(f)$. For case 2 and $l = 0$ we have $x^* = d_0$ and $\underline{z}^* = 0$ so $(x^*, \underline{z}^*) \in \text{epi}(f)$. For case 3 we have $x^* = (d_{l-1} + d_l)/2$ and $\underline{z}^* = (m_{l-1} d_{l-1} + c_{l-1} + b_l + m_l d_l + c_l + b_{l+1})/2 \geq (m_l(d_{l-1} + d_l)/2 + c_l + b_l$ so $(x^*, \underline{z}^*) \in \text{epi}(f)$. For case 4 we have $x^* = (d_{l-1} + d_l)/2$ and $\underline{z}^* = (m_{l-1} d_{l-1} + c_{l-1} + b_{l-1} + m_l d_l + c_l + b_l)/2$, but $(d_k, m_k d_k + c_k + b_k) \in \text{epi}(f)$ so $(x^*, \underline{z}^*) \in \text{conv}(\text{epi}(f))$. □

EC.3. Additional Computational Results for Univariate Continuous Functions

In this section we extend the computational results of Section 5.1 by considering instances with $K = 4$, 16 and 32. Tables EC.1, EC.2 and EC.3 present the same statistics as Table 3 for this set of instances. We note that for these values of K the average integrality GAPS are 4%, 6% and 6% respectively.

model	min	**avg**	max	std
MC	26	**58**	100	17.7
DLog	16	**37**	62	10.8
Inc	12	**37**	100	16.0
DCC	16	**36**	60	10.2
Log	13	**36**	61	10.0
CC	11	**25**	43	6.7
SOS2	0	**0**	0	0.0

(a) GAP closed at root node by CPLEX. [%]

model	min	**avg**	max	std
SOS2	0	**2.8**	10	2.8
Log	0	**5.3**	10	5.3
DLog	0	**5.7**	10	5.7
DCC	0	**5.9**	10	5.9
CC	0	**7.1**	10	7.1
MC	0	**9.4**	20	9.4
Inc	0	**12.4**	20	12.4

(b) LP relaxation solve time. [ms]

model	min	**avg**	max	std
MC	0	**234**	891	216
Inc	1	**357**	2081	365
DLog	22	**504**	2677	529
Log	14	**587**	3569	617
DCC	10	**798**	5960	897
CC	30	**964**	8938	1139
SOS2	220	**1974**	13434	1833

(c) Branch-and-bound nodes processed.

Table EC.1 Solve characteristics for univariate continuous functions and $K=4$.

model	min	**avg**	max	std
MC	10.9	**26**	53	7.2
DCC	7.1	**17**	48	5.9
DLog	2.0	**17**	51	8.1
Log	2.0	**17**	51	7.7
Inc	2.6	**14**	35	5.9
CC	5.6	**10**	21	2.8
SOS2	0.0	**0**	0	0.0

(a) GAP closed at root node by CPLEX. [%]

model	min	**avg**	max	std
SOS2	0	**5.9**	10	5.9
Log	0	**15.8**	20	15.8
CC	10	**23.3**	40	23.3
DLog	10	**27.7**	40	27.7
DCC	10	**29.9**	40	29.9
MC	50	**89.6**	120	89.5
Inc	90	**139.0**	180	138.9

(b) LP relaxation solve time. [ms]

model	min	**avg**	max	std
MC	52	**2809**	27890	4392
DLog	44	**4129**	19900	3978
Log	45	**4428**	23921	4167
Inc	204	**5139**	25162	4118
CC	245	**28895**	241524	38696
SOS2	1487	**98050**	959307	155930
DCC	461	**302134**	2345087	461223

(c) Branch-and-bound nodes processed.

Table EC.2 Solve characteristics for univariate continuous functions and $K=16$.

model	min	**avg**	max	std
MC	9.5	**18.8**	31.4	4.8
Log	1.7	**15.0**	32.3	6.7
DCC	5.8	**13.4**	25.4	3.7
Inc	1.8	**9.5**	24.4	4.3
CC	2.0	**5.6**	9.5	1.5
DLog	0.1	**1.8**	11.6	1.9
SOS2	0.0	**0.0**	0.0	0.0

(a) GAP closed at root node by CPLEX. [%]

model	min	**avg**	max	std
SOS2	0	**12**	20	11
Log	10	**30**	40	30
CC	20	**39**	60	39
DLog	40	**60**	70	60
DCC	60	**93**	110	93
MC	230	**418**	600	418
Inc	410	**534**	670	534

(b) LP relaxation solve time. [ms]

model	min	**avg**	max	std
DLog	382	**4776**	27375	4926
Log	276	**5287**	25797	5505
Inc	964	**8196**	40352	7315
MC	471	**28855**	146197	37678
CC	1762	**80224**	505999	103995
SOS2	2752	**471156**	4707352	943424
DCC	5097	**916227**	1485910	389175

(c) Branch-and-bound nodes processed.

Table EC.3 Solve characteristics for univariate continuous functions and $K = 32$.

EC.4. Example for Lower Semicontinuous Functions

The function given in Figure 3(a) can be described as

$$g(x) := \begin{cases} 1.5x+1 & x\in[0,1) \\ 2 & x\in[2,2] \\ -1.5x+6 & x\in(2,4] \\ 2x-7 & x\in(4,5] \end{cases} \tag{EC.3}$$

and characterization (17) of its epigraph is given by

$$\begin{aligned}\operatorname{epi}(g) = \{(0,r) : r\geq 0\} + \Big(&\operatorname{conv}\big(\{(0,1),(2,4)\}\big)\cup\operatorname{conv}\big(\{(2,2)\}\big)\\ &\cup\operatorname{conv}\big(\{(2,3),(4,0)\}\big)\cup\operatorname{conv}\big(\{(4,1),(5,3)\}\big)\Big).\end{aligned}$$

We now describe the formulations in Section 3 for the function defined in (EC.3). DCC is given by

$$0\lambda_{[0,2),0} + 2\left(\lambda_{[0,2),2}+\lambda_{[2,2],2}+\lambda_{(2,4],2}\right) + 4\left(\lambda_{(2,4],4}+\lambda_{(4,5],4}\right) + 5\lambda_{(4,5],5} = x$$

$$1\lambda_{[0,2),0} + 4\lambda_{[0,2),2} + 2\lambda_{[2,2],2} + 3\lambda_{(2,4],2} + 0\lambda_{(2,4],4} + 1\lambda_{(4,5],4} + 3\lambda_{(4,5],5} \leq z$$

$$\lambda_{[0,2),0},\, \lambda_{[0,2),2},\, \lambda_{[2,2],2},\, \lambda_{(2,4],2},\, \lambda_{(2,4],4},\, \lambda_{(4,5],4},\, \lambda_{(4,5],5} \geq 0$$

$$\lambda_{[0,2),0} + \lambda_{[0,2),2} = y_{[0,2)}, \quad \lambda_{[2,2],2} = y_{[2,2]}, \quad \lambda_{(2,4],2} + \lambda_{(2,4],4} = y_{(2,4]}, \quad \lambda_{(4,5],4} + \lambda_{(4,5],5} = y_{(4,5]}$$

$$y_{[0,2)} + y_{[2,2]} + y_{(2,4]} + y_{(4,5]} = 1, \quad y_{[0,2)},\, y_{[2,2]},\, y_{(2,4]},\, y_{(4,5]} \in \{0,1\}.$$

For $B([0,2)) = (0,0)^T$, $B([2,2]) = (0,1)^T$, $B((2,4]) = (1,1)^T$, $B((4,5]) = (1,0)^T$ DLog is given by

$$0\lambda_{[0,2),0} + 2\left(\lambda_{[0,2),2} + \lambda_{[2,2],2} + \lambda_{(2,4],2}\right) + 4\left(\lambda_{(2,4],4} + \lambda_{(4,5],4}\right) + 5\lambda_{(4,5],5} = x$$

$$1\lambda_{[0,2),0} + 4\lambda_{[0,2),2} + 2\lambda_{[2,2],2} + 3\lambda_{(2,4],2} + 0\lambda_{(2,4],4} + 1\lambda_{(4,5],4} + 3\lambda_{(4,5],5} \leq z$$

$$\lambda_{[0,2),0},\, \lambda_{[0,2),2},\, \lambda_{[2,2],2},\, \lambda_{(2,4],2},\, \lambda_{(2,4],4},\, \lambda_{(4,5],4},\, \lambda_{(4,5],5} \geq 0$$

$$\lambda_{(2,4],2} + \lambda_{(2,4],4} + \lambda_{(4,5],4} + \lambda_{(4,5],5} \leq y_1, \quad \lambda_{[0,2),0} + \lambda_{[0,2),2} + \lambda_{[2,2],2} \leq (1 - y_1)$$

$$\lambda_{[2,2],2} + \lambda_{(2,4],2} + \lambda_{(2,4],4} \leq y_2, \quad \lambda_{[0,2),0} + \lambda_{[0,2),2} + \lambda_{(4,5],4} + \lambda_{(4,5],5} \leq (1 - y_2).$$

$$y_1,\, y_2 \in \{0,1\}.$$

MC is given by

$$x^{[0,2)} + x^{[2,2]} + x^{(2,4]} + x^{(4,5]} = x$$

$$\left(1.5x^{[0,2)} + 1y_{[0,2)}\right) + \left(0x^{[2,2]} + 2y_{[2,2]}\right) + \left(-1.5x^{(2,4]} + 6y_{(2,4]}\right) + \left(2x^{(4,5]} - 7y_{[4,5]}\right) \leq z$$

$$0y_{[0,2)} \leq x^{[0,2)} \leq y_{[0,2)}, \quad 2y_{[2,2]} \leq x^{[2,2]} \leq 2y_{[2,2]}, \quad 2y_{(2,4]} \leq x^{(2,4]} \leq 4y_{(2,4]}, \quad 4y_{(4,5]} \leq x^{(4,5]} \leq 5y_{(4,5]}$$

$$y_{[0,2)} + y_{[2,2]} + y_{(2,4]} + y_{(4,5]} = 1, \quad y_{[0,2)},\, y_{[2,2]},\, y_{(2,4]},\, y_{(4,5]} \in \{0,1\}.$$

For g defined in (EC.3) $\operatorname{epi}(g) = C_1^+ + \left(\bigcup_{k=1}^K \operatorname{conv}\left(\{(d_{k-1}, f_{k-1}), (d_k, f_k)\}\right)\right)$ for $K = 6,\ d_0 = 0,\ d_1 = d_2 = d_3 = 2,\ d_4 = d_5 = 4,\ d_6 = 5,\ f_0 = g(0) = 1,\ f_1 = g^-(2) = 4,\ f_2 = g(2) = 2,\ f_3 = g^+(2) = 3,\ f_4 = g(4) = 0,\ f_5 = g^+(4) = 1$ and $f_6 = g(5) = 3$. CC Dup is given by

$$0\lambda_0 + 2(\lambda_1 + \lambda_2 + \lambda_3) + 4(\lambda_4 + \lambda_5) + 5\lambda_6 = x, \quad 1\lambda_0 + 4\lambda_1 + 2\lambda_2 + 3\lambda_3 + 0\lambda_4 + 1\lambda_5 + 3\lambda_6 \leq z$$

$$\lambda_0,\, \lambda_1,\, \lambda_2,\, \lambda_3,\, \lambda_4,\, \lambda_5,\, \lambda_6 \geq 0, \quad \lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 + \lambda_6 = 1$$

$$\lambda_0 \leq y_0, \quad \lambda_1 \leq y_1 + y_2, \quad \lambda_2 \leq y_2 + y_3, \quad \lambda_3 \leq y_3 + y_4, \quad \lambda_4 \leq y_4 + y_5, \quad \lambda_5 \leq y_5 + y_6, \quad \lambda_6 \leq y_6$$

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 = 1, \quad y \in \{0,1\}^6.$$

For $G^1 = (0,0,0)^T$, $G^2 = (1,0,0)^T$, $G^3 = (1,1,0)^T$, $G^4 = (0,1,0)^T$, $G^5 = (0,1,1)^T$, $G^6 = (1,1,1)^T$ Log Dup is given by

$$0\lambda_0 + 2(\lambda_1+\lambda_2+\lambda_3) + 4(\lambda_4+\lambda_5) + 5\lambda_6 = x, \quad 1\lambda_0 + 4\lambda_1 + 2\lambda_2 + 3\lambda_3 + 0\lambda_4 + 1\lambda_5 + 3\lambda_6 \leq z$$

$$\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6 \geq 0, \quad \lambda_0+\lambda_1+\lambda_2+\lambda_3+\lambda_4+\lambda_5+\lambda_6 = 1$$

$$\lambda_2+\lambda_6 \leq y_1, \quad \lambda_0+\lambda_4 \leq (1-y_1), \quad \lambda_3+\lambda_4+\lambda_5+\lambda_6 \leq y_2, \quad \lambda_0+\lambda_1 \leq (1-y_2)$$

$$\lambda_5+\lambda_6 \leq y_3, \quad \lambda_0+\lambda_1+\lambda_2+\lambda_3 \leq (1-y_3), \quad y \in \{0,1\}^3.$$

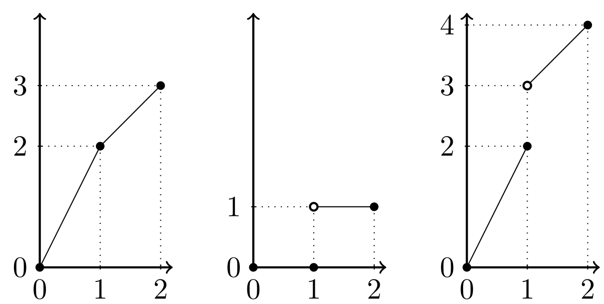
Inc Dup is given by

$$0 + 2\delta_1 + 0\delta_2 + 0\delta_3 + 2\delta_4 + 0\delta_5 + 1\delta_6 = x, \quad 1 + 3\delta_1 - 2\delta_2 + 1\delta_3 - 3\delta_4 + 1\delta_5 + 2\delta_6 \leq z$$

$$y_1 \leq \delta_1 \leq 1, \quad y_2 \leq \delta_2 \leq y_1, \quad y_3 \leq \delta_3 \leq y_2, \quad y_4 \leq \delta_4 \leq y_3, \quad y_5 \leq \delta_5 \leq y_4, \quad 0 \leq \delta_6 \leq y_5, \quad y \in \{0,1\}^5.$$

Finally, we describe formulations with the coordination technique for the function $\tilde{g} = \tilde{g}_C + \tilde{g}_J$ in Figure EC.1. This function can be described in form (21) for $K=2$, $d_0=0$, $d_1=1$, $d_2=2$, $m_1=2$, $m_2=1$, $c_1=0$, $c_2=1$, $b_1=0$ and $b_2=1$, which yields

$$\tilde{g}(x) := \begin{cases} 2x & x \in [0,1] \\ x+2 & x \in (1,2]. \end{cases} \tag{EC.4}$$



(a) $\tilde{g}_C$. (b) $\tilde{g}_J$. (c) $\tilde{g}$.

Figure EC.1 Decomposition of fixed charged lower semicontinuous piecewise linear function.

For $\tilde{g}$, CC Coord is given by

$$0\lambda_0 + 1\lambda_1 + 2\lambda_2 = x, \quad 0w_1 + 1w_2 + 0\lambda_0 + 2\lambda_1 + 3\lambda_2 \leq z, \quad \lambda_0+\lambda_1+\lambda_2 = 1$$

$$\lambda_0 \leq w_1, \quad \lambda_1 \leq w_1 + w_2, \quad \lambda_2 \leq w_2, \quad \lambda_0, \lambda_1, \lambda_2 \geq 0, \quad w_1 + w_2 = 1, \quad w_1, w_2 \in \{0,1\}$$

and Inc Coord is given by

$$\delta_1 + \delta_2 = x, \quad 2\delta_1 + 1\delta_2 + w_1 \leq z, \quad w_1 \leq \delta_1 \leq 1, \quad 0 \leq \delta_2 \leq w_1, \quad w_1 \in \{0,1\}.$$