A note on :"A Superior Representation Method for Piecewise Linear Functions" by Li, Lu, Huang and Hu

Juan Pablo Vielma, Shabbir Ahmed and George Nemhauser
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive, NW, Atlanta, GA 30332-0205, USA, {jvielma@isye.gatech.edu, sahmed@isye.gatech.edu, george.nemhauser@isye.gatech.edu}

This paper shows that two Mixed Integer Linear Programming (MILP) formulations for piecewise linear functions introduced by Li et al. (2008) are both theoretically and computationally inferior to standard MILP formulations for piecewise linear functions.

Key words: Mathematics:Piecewise linear; Programming:Integer;
History:

---

1. Introduction

Two new Mixed Integer Linear Programming (MILP) formulations for modeling a univariate piecewise linear function f were introduced in Li et al. (2008). The first formulation (given by (1)–(3) in Li et al.) uses "Big-M" type constraints, so we denote it by *LiBigM*. The second formulation (given by (23)–(33) in Li et al.) uses a number of binary variables that is logarithmic in the number of segments in which f is affine, so we denote it by *LiLog*. Based on computational results that show that LiLog outperforms LiBigM, Li et al. declare LiLog to be superior to other MILP formulations for piecewise linear functions. In this paper we show that LiBigM and LiLog are both theoretically and computationally inferior to standard MILP formulations for piecewise linear functions.

In Section 2 we show that both formulations from Li et al. are theoretically inferior to essentially every standard MILP formulation for piecewise linear functions. In Section 3 we present results of computational experiments that compare the formulations from Li et al. to two other standard formulations.

2. Strength of Formulations

An MILP formulation for a univariate piecewise linear function $f : D \to \mathbb{R}$ is $Q := \{(x, y, \lambda, \mu) \in P : \mu \in \{0,1\}^q\}$ such that P is a polyhedron and $\operatorname{proj}_{(x,y)}(Q) = \{(x, y) : f(x) = y\}$, where $\operatorname{proj}_{(x,y)}(\cdot)$ is the projection onto the (x, y) variables. An MILP formulation Q is said to be *sharp* (Jeroslow and Lowe, 1984) if its linear programming (LP) relaxation P is such that $y \geq g(x)$ for all $(x, y) \in \operatorname{proj}_{(x,y)}(P)$, where $g := \operatorname{convenv}_D(f)$ is the lower convex envelope of f over D. Sharp formulations provide the best possible LP relaxation bounds so the sharpness property is crucial for the efficient solution of these problems using branch-and-bound. An MILP formulation is *locally ideal* (Padberg, 2000) if P has integral extreme points. The locally ideal property can provide an additional advantage because it implies the sharpness property (e.g. Vielma et al., 2008). The sharpness property is shared by essentially every standard MILP formulation for piecewise linear functions (see Vielma et al.) and most of them are also locally ideal. We now show that neither LiBigM nor LiLog is sharp.

For LiBigM we use the piecewise linear function $f : [0, 4] \to \mathbb{R}$ defined by

$$f(x) := \begin{cases} 3x & x \in [0,1] \\ 2x+1 & x \in [1,4], \end{cases} \tag{1}$$

for which $g(1) = 2.25$ for $g = \operatorname{convenv}_D(f)$. LiBigM for this function is

$$\begin{gathered} -4(1-\lambda_0) \leq x \leq 1 + 4(1-\lambda_0) \\ 1 - 4(1-\lambda_1) \leq x \leq 2 + 4(1-\lambda_1) \\ 3x - 12(1-\lambda_0) \leq y \leq 3x + 12(1-\lambda_0) \\ 2x + 1 - 12(1-\lambda_1) \leq y \leq 2x + 1 + 12(1-\lambda_1) \\ \lambda_0 + \lambda_1 = 1, \quad \lambda_0, \lambda_1 \in \{0,1\} \end{gathered}$$

which has $\lambda_0 = \lambda_1 = 1/2$, $x = 1$, and $y = -3 < 2.25$ as a feasible solution to its LP relaxation.

For LiLog we use the piecewise linear function $f : [0, 4] \to \mathbb{R}$ defined by

$$f(x) := \begin{cases} 4x & x \in [0,1] \\ 3x+1 & x \in [1,2] \\ 2x+3 & x \in [2,3] \\ x+6 & x \in [3,4], \end{cases}$$

for which $g(1) = 2.5$ for $g = \text{convenv}_D(f)$. LiLog for this function is

$$\begin{aligned}
r_1 + 2r_2 + 3r_3 \le x \ &\le r_0 + 2r_1 + 3r_2 + 4r_3 \\
r_1 + 3r_2 + 6r_3 + 4w_0 + 3w_1 + 2w_2 + w_3 &= y \\
r_0 + r_1 + r_2 + r_3 &= 1 \\
r_1 + r_2 + 2r_3 + z_1 + z_2 &= 0 \\
-u'_1 \le z_1 &\le u'_1 \\
-u'_2 \le z_2 &\le u'_2 \\
r_0 - r_1 + r_2 - r_3 - (1 - u'_1) \le z_1 &\le r_0 - r_1 + r_2 - r_3 + (1 - u'_1) \\
r_0 + r_1 - r_2 - r_3 - (1 - u'_2) \le z_2 &\le r_0 + r_1 - r_2 - r_3 + (1 - u'_2) \\
w_0 + w_1 + w_2 + w_3 &= x \\
w_1 + w_2 + 2w_3 + \delta_1 + \delta_2 &= 0 \\
-4u'_1 \le \delta_1 &\le 4u'_1 \\
-4u'_2 \le \delta_2 &\le 4u'_2 \\
w_0 - w_1 + w_2 - w_3 - 4(1 - u'_1) \le \delta_1 &\le w_0 - w_1 + w_2 - w_3 + 4(1 - u'_1) \\
w_0 + w_1 - w_2 - w_3 - 4(1 - u'_2) \le \delta_2 &\le w_0 + w_1 - w_2 - w_3 + 4(1 - u'_2) \\
u'_1 + 2u'_2 &\le 3 \\
r_0, r_1, r_2, r_3, w_0, w_1, w_2, w_3 &\ge 0 \\
&u'_1, u'_2 \in \{0, 1\}
\end{aligned}$$

which has $x = 1$, $r_0 = r_1 = 0.5$, $r_2 = r_3 = 0$, $u'_1 = 0.5$, $u'_2 = 0$, $w_0 = w_1 = w_2 = 0$, $w_3 = 1$, $z_1 = -0.5$, $z_2 = 0$, $\delta_1 = -2.0$, $\delta_2 = 0$ and $y = 1.5 < 2.5$ as a feasible solution to its LP relaxation.

3. Computational Results

We now present a computational comparison between LiBigM, LiLog and two standard MILP formulations. Most standard MILP formulations for piecewise linear functions are studied in Vielma et al. From these formulations we select the so called *Convex Combination Model* and *Logarithmic Convex Combination Model* as a representative sample. The Convex

Combination Model appears as early as Dantzig (1960) and is included in many textbooks (Dantzig, 1963; Garfinkel and Nemhauser, 1972; Nemhauser and Wolsey, 1988). Although it is a sharp formulation, it is the only formulation studied in Vielma et al. that is not locally ideal and it has one of the worst computational performances. Hence it is an example of a classical, but relatively weak formulation. The Logarithmic Convex Combination Model is a sharp and locally ideal formulation introduced in Vielma and Nemhauser (2008a,b) that has a number of binary variables and constraints that is logarithmic in the number of segments in which the modeled function is affine. It has one of the best computational performances in Vielma et al. and hence is an example of a state of the art formulation. We denote the Convex Combination and Logarithmic Convex Combination Models by *CC* and *Log* respectively.

The first set of instances are from Li et al. and consists of a series of Mixed Integer Nonlinear Programming (MINLP) problems that were obtained by linearizing some Nonlinear Nonconvex Programming (NNP) problems. These MINLPs were obtained by replacing the nonconvex portions of the NNPs by univariate piecewise linear approximations and correspond to Examples 1 and 2 in Li et al. To solve these instances we used Bonmin 1.0.1 (Bonami et al., 2005) with CPLEX 11 (ILOG, 2008) as an MILP subsolver on a 2.4GHz workstation with 2GB of RAM. We selected the Hybrid solver from Bonmin because most of the time it significantly outperforms the other Bonmin solvers. Table 1 shows the solve times in seconds for the different instances, which are identified according to their parameters (e.g. Example 1 in Li et al. has three possible sets of parameter that we identify as 1a, 1b and 1c) and the resolution of the piecewise linear approximations (e.g. Example 2 in Li et al. included approximations with piecewise linear functions with 32 and 64 segments).

	Example 1a		Example 1b			Example 1c			Example 2	
segments	64	256	64	128	256	64	128	256	32	64
Log	1.1	3.3	0.2	0.4	0.7	0.3	0.4	0.7	6.1	11.2
CC	4.8	12.1	1.0	1.6	3.2	1.8	2.1	5.7	24.1	59.5
LiLog	4.3	23.5	1.1	2.2	5.4	1.2	2.5	5.6	116.0	129.2
LiBigM	1.9	19.2	1.0	2.5	17.5	1.0	4.5	15.8	214.0	306.0

Table 1: Solve times for MINLPs using Bonmin's hybrid algorithm [s].

We see that LiLog is rarely faster than CC and is always at least four times slower than Log. In fact, for the instances from Example 2 the solve times of both LiLog and LiBigM are more than twice the time of CC and over an order of magnitude the time of Log.

The second set of instances from Li et al. are MILPs resulting from problems with univariate piecewise linear functions. These instances include Example 4 and a variant of Example 1 from Li et al constructed by replacing every nonlinearity (convex and nonconvex) of the original NNP with a piecewise linear approximation. We note that for this variant we did not treat the convex nonlinearities specially so that we could assess the performance of the different formulations even in the case in which some of the piecewise linear functions can actually be modeled as LPs. These instances were solved using CPLEX 11. Table 2 shows the results for these instances in the same format as Table 1.

	Example 1a		Example 1b			Example 1c			Example 4		
segments	64	256	64	128	256	64	128	256	32	64	128
Log	0.03	0.07	0.04	0.03	0.10	0.02	0.03	0.11	0.10	0.24	0.47
CC	0.08	0.57	0.38	0.41	1.38	0.30	0.62	1.41	0.58	1.79	3.77
LiLog	0.45	2.86	0.37	1.47	4.60	0.34	1.41	8.96	6.21	34.09	260.68
LiBigM	1.35	73.34	1.40	8.75	66.90	1.16	7.60	46.78	11.22	56.35	428.06

Table 2: Solve times for MILPs from Li et al. (2008) [s].

The results are very similar to those for the first set and agree with the fact that BigM and LiLog are theoretically inferior to standard formulation for piecewise linear functions.

The final set of instances consists of the transportation problems with piecewise linear cost functions studied in Vielma et al. These instances consider univariate piecewise linear functions that are affine in K segments for $K \in \{4, 8, 16, 32\}$ and include 100 randomly generated instances for each K. We again used CPLEX 11 as an MILP solver and Table 3 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 of 10,000 seconds was reached. We note that LiBigM was not considered for $K = 16$ and 32 because it had already failed too many times for $K = 8$ and that the statistics for LiLog with $K = 32$ are marked with a dash (-) because it failed in every single instance.

We again see that the theoretically inferior formulations LiBigM and LiLog are significantly slower than CC and Log. In addition, the results from Table 3 can be compared with the results in Vielma et al. to see that both LiBigM and LiLog are significantly slower than each of the six formulation tested in Vielma et al. for this set of instances.

	min	**avg**	max	std	fail
Log	0.2	**2.1**	12	2.3	0
CC	0.3	**4.6**	23	4.3	0
LiLog	10.0	**25.5**	124	15.1	0
BigM	17.4	**652.2**	9951	1245.2	0

(a) 4 segments.

	min	**avg**	max	std	fail
Log	0.6	**12**	84	11	0
CC	2.6	**81**	570	97	0
LiLog	88.9	**2702**	10000	3024	11
BigM	259.8	**6380**	10000	3742	44

(b) 8 segments.

	min	**avg**	max	std	fail
Log	0.5	**24**	96	18	0
CC	3.9	**351**	3691	517	0
LiLog	848.0	**9863**	10000	982	97

(c) 16 segments.

	min	**avg**	max	std	fail
Log	2.5	**43**	194	39	0
CC	67.5	**1938**	10000	2560	4
LiLog	-	**-**	-	-	100

(d) 32 segments.

Table 3: Solve times for univariate continuous functions [s].

We finally note that these negative results only concern formulations LiBigM and LiLog that Li et al. use to model piecewise linear functions. The results in Table 4 from Li et al. suggest that their main ideas could be useful for other problems such as the unique selection over a finite set of choices.

Acknowledgments

This research has been supported by NSF grants CMMI-0522485 and CMMI-0758234 and AFOSR grant FA9550-07-1-0177.

References

Bonami, P., A. Waechter, L. T. Biegler, A. R. Conn, G. Cornuejols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya. 2005. An algorithmic framework for convex mixed integer nonlinear programs. IBM Research Report RC23771, IBM, Yorktown Heights, NY.

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.

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

ILOG. 2008. *ILOG CPLEX User's Manual*. ILOG, S.A.

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

Li, H.-L., H.-C. Lu, C.-H. Huang, N.-Z. Hu. 2008. A superior representation method for piecewise linear functions. *Informs Journal on Computing* **To appear, DOI: 10.1287/ijoc.1080.0294**.

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. *Operations Research Letters* **27** 1–5.

Vielma, J. P., S. Ahmed, G. L. Nemhauser. 2008. Mixed-integer models for nonseparable piecewise linear optimization: unifying framework and extensions. *Operations Research* **To Appear**.

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. *Mathematical Programming* **Under second review**.