SAND2012-0505P

Submitted to *Operations Research*
manuscript OPRE-2011-01-001

Base-2 Expansions for Linearizing Products of Functions of Discrete Variables

Warren P. Adams
Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975, wadams@clemson.edu

Stephen M. Henry
System Readiness and Sustainment Technologies Group, Sandia National Laboratories, Albuquerque, New Mexico 87123, smhenry@sandia.gov

This paper presents an approach for representing functions of discrete variables, and their products, using logarithmic numbers of binary variables. Given a univariate function whose domain consists of n distinct values, it begins by employing a base-2 expansion to express the function in terms of the ceiling of $\log_2 n$ binary and n continuous variables, using linear restrictions to equate the functional values with the possible binary realizations. The representation of the product of such a function with a nonnegative variable is handled via an appropriate scaling of the linear restrictions. Products of m functions are treated in an inductive manner from $i=2$ to m, where each step i uses such a scaling to express the product of function i and a nonnegative variable denoting a translated version of the product of functions 1 through $i-1$ as a newly-defined variable. The resulting representations, both in terms of one function and many, are important for reformulating general discrete variables as binary, and also for linearizing mixed-integer generalized geometric and discrete nonlinear programs, where it is desired to economize on the number of binary variables. The approach provides insight into, improves upon, and subsumes related linearization methods for products of functions of discrete variables.

Key words: Programming; integer; nonlinear; theory.

History: This paper was first submitted on January 3, 2011 and has been with the authors for 2 months for one revision.

---

1. Introduction

Consider a discrete variable x that can realize values in the finite set $S=\{\theta_1,\theta_2,\ldots,\theta_n\}$. It is well known that x can be expressed in terms of n binary variables $\boldsymbol{\lambda}^T=(\lambda_1,\lambda_2,\ldots,\lambda_n)$ as

$$x=\sum_{j=1}^{n}\theta_j\lambda_j,\ \boldsymbol{\lambda}\in\Lambda, \tag{1}$$

where

$$\Lambda \equiv \left\{ \boldsymbol{\lambda} \in \mathbb{R}^n : \sum_{j=1}^{n} \lambda_j = 1,\ \lambda_j \text{ binary for } j = 1, \ldots, n \right\}. \tag{2}$$

Moreover, given that x is an integer with $\theta_j = \theta_{j-1} + 1$ for $j = 2, \ldots, n$, then x can be alternately defined as in Watters (1967) by

$$x = \theta_1 + \sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k,\ x \le \theta_n,\ u_k \text{ binary for } k = 1, \ldots, \lceil \log_2 n \rceil. \tag{3}$$

Of course, if $\lceil \log_2 n \rceil = \log_2 n$, then the inequality $x \le \theta_n$ of (3) is not needed. (Throughout this paper, we find it convenient to denote sums from 1 to n using the index j and sums from 1 to $\lceil \log_2 n \rceil$ using the index k.)

An obvious difference between (1) and (3) is that the former requires n binary variables whereas the latter uses only $\lceil \log_2 n \rceil$. In this study, we represent *functions* of discrete variables in terms of logarithmic numbers of binary variables, and use these representations to linearize products of such functions. A recent work of Li and Lu (2009) has contributed two such linearizations by defining auxiliary continuous variables and linear constraints. The methods vary in their construction. This raises the following two-part question. Given a discrete variable x that can realize values in some arbitrary set S having $|S| = n$, how can x be most economically represented, and how can such a representation be used to linearize products of discrete functions?

We use a simple observation relative to the unit hypercube to address this question so as to efficiently represent x and any associated function $f(x)$, and ultimately to represent products of such functions. As a consequence, we are able to improve upon the contributions of Li and Lu (2009) relative to the linearization of monomial terms of discrete variables, as well as to mixed-integer generalized geometric programs. This paper is in the spirit of work in Vielma and Nemhauser (2011), which presents an interesting study on the use of logarithmic numbers of binary variables to model disjunctive constraints, focusing on SOS1 and SOS2 type restrictions.

Applications for functions of discrete variables naturally arise in a broad range of fields, including environmentally benign solvent design (Sinha et al. 1999), molecular design of freon alternatives (Sahinidis and Tawarmalani 2000), pooling problems for chemical and wastewater treatment

(Meyer and Floudas 2006), optimization of heat exchange networks (Bergamini et al. 2007), nonconvex portfolio optimization (Kallrath 2003), and digital circuit optimization (Boyd et al. 2005), to name a few. The paper by Floudas and Gounaris (2009) gives an excellent overview of recent advances and applications for a variety of problems involving functions of discrete variables. Applications also arise involving *products* of discrete variables, or functions thereof. For example, Harjunkoski et al. (1999) study trim loss minimization in the paper-converting industry when slicing large paper spools into smaller output pieces to meet customer specifications. Here, nonnegative integer variables are used to represent the number of times a particular cutting pattern is selected, as well as the number of each type output that is produced by a specific cutting pattern. Products of these variables represent the total numbers of outputs, and ensure customer demands are met.

2. Base-2 Representations of Discrete Variables and Functions

In this section, we represent a discrete variable $x \in S = \{\theta_1, \theta_2, \ldots, \theta_n\}$ in terms of $\lceil \log_2 n \rceil$ binary variables, n nonnegative continuous variables, and $\lceil \log_2 n \rceil + 1$ linear equality restrictions. The representation is then shown to extend to functions of this variable, as well as to the product of any such function with a nonnegative variable. The study relies on the following elementary observation, stated without proof due to its simplicity.

Observation

Given any positive integer p, a binary vector $\boldsymbol{u} \in \mathbb{R}^p$ can be represented as a convex combination of a select subset of $n \le 2^p$ distinct extreme points of the unit hypercube in $\mathbb{R}^p$ if and only if the vector $\boldsymbol{u}$ is itself one of the selected extreme points, with a single convex multiplier equaling 1, and the remaining $n-1$ multipliers equaling 0.

For our purposes, a useful implementation of this observation is the following. Consider any n distinct extreme points $\boldsymbol{v}_j$, $j \in \{1, \ldots, n\}$, of the unit hypercube in $\mathbb{R}^{\lceil \log_2 n \rceil}$. For simplicity, we henceforth define these extreme points so that vector $\boldsymbol{v}_j \in \mathbb{R}^{\lceil \log_2 n \rceil}$ is the base-2 expansion of the number $j-1$, where entry i corresponds to the value 2^{i-1}. Let $\boldsymbol{\lambda} \in \mathbb{R}^n$ serve as convex multipliers of these points $\boldsymbol{v}_j$. Then the observation gives us, with $p = \lceil \log_2 n \rceil$, that $\boldsymbol{\lambda} \in \Lambda$ of (2) if and only if there exists a vector $\boldsymbol{u} \in \mathbb{R}^{\lceil \log_2 n \rceil}$ so that $(\boldsymbol{u}, \boldsymbol{\lambda}) \in \Lambda'$, where

$$\Lambda' \equiv \left\{ (\boldsymbol{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n : \sum_{j=1}^{n} \lambda_j = 1, \; \sum_{j=1}^{n} \boldsymbol{v}_j \lambda_j = \boldsymbol{u}, \; \boldsymbol{u} \text{ binary}, \; \boldsymbol{\lambda} \geq \mathbf{0} \right\}. \tag{4}$$

Consequently, (4) provides a mechanism for replacing the restrictions $\boldsymbol{\lambda} \in \Lambda$ of (2) in n binary variables with $(\boldsymbol{u}, \boldsymbol{\lambda}) \in \Lambda'$ in $\lceil \log_2 n \rceil$ binary variables. This gives us that x described in (1) and (2) can be expressed with $\lceil \log_2 n \rceil$ binary variables $\boldsymbol{u}$, n nonnegative continuous variables $\boldsymbol{\lambda}$, and $\lceil \log_2 n \rceil + 1$ equality constraints from (4) in $\boldsymbol{\lambda}$ and $\boldsymbol{u}$ as

$$x = \sum_{j=1}^{n} \theta_j \lambda_j, \; (\boldsymbol{u}, \boldsymbol{\lambda}) \in \Lambda'. \tag{5}$$

The region Λ' of (4) possesses an interesting property, and provides insight into related sets. From a polyhedral perspective, Λ' is *locally ideal* in that the polytope obtained by removing the $\boldsymbol{u}$ binary restrictions has $\boldsymbol{u}$ (and $\boldsymbol{\lambda}$) binary at all extreme points. This is readily seen since the set defined by $\sum_{j=1}^{n} \lambda_j = 1$ and $\boldsymbol{\lambda} \geq \mathbf{0}$ has each extreme point having a single λ_j equaling 1 and the rest equaling 0, and since the constraints $\sum_{j=1}^{n} \boldsymbol{v}_j \lambda_j = \boldsymbol{u}$ consequently serve only to fix $\boldsymbol{u}$ to some binary $\boldsymbol{v}_j$. Λ' also relates to a form found in Vielma and Nemhauser (2011), and used in the context of piecewise linear models within Vielma et al. (2010b). This set, which Vielma et al. (2010b) show to be locally ideal, is

$$\Lambda'' \equiv \left\{ \begin{array}{l} (\boldsymbol{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n : \\ \qquad \displaystyle\sum_{j=1}^{n} \lambda_j = 1, \; \sum_{j=1}^{n} \boldsymbol{v}_j \lambda_j \leq \boldsymbol{u}, \; \sum_{j=1}^{n} (\mathbf{1} - \boldsymbol{v}_j) \lambda_j \leq \mathbf{1} - \boldsymbol{u}, \; \boldsymbol{u} \text{ binary}, \; \boldsymbol{\lambda} \geq \mathbf{0} \end{array} \right\}.$$

Subtracting $\sum_{j=1}^{n} \lambda_j = 1$ from each constraint in $\sum_{j=1}^{n} (\mathbf{1} - \boldsymbol{v}_j) \lambda_j \leq \mathbf{1} - \boldsymbol{u}$ gives us that $\sum_{j=1}^{n} \boldsymbol{v}_j \lambda_j \geq \boldsymbol{u}$ so that Λ' of (4) and Λ'' are equivalent, alternately establishing Λ' as locally ideal. Notably, Λ' can be viewed as an improvement over Λ'' as it contains around half the number of constraints. (The paper of Vielma and Nemhauser (2011) has the equality of Λ'' relaxed to $\leq$, but a similar ideal argument holds for the inequality case. The equality version is considered in Vielma et al. (2010b).)

It is instructive to note cases of S for which (5) can be reduced to the size of (3), and contain no $\boldsymbol{\lambda}$ variables. Define the $(\lceil \log_2 n \rceil + 1) \times n$ matrix V whose j^{th} column is given by $\begin{bmatrix} 1 \\ \boldsymbol{v}_j \end{bmatrix}$ so that the equations of Λ' can be written as

$$V\boldsymbol{\lambda} = \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}. \tag{6}$$

Suppose that the vector $\boldsymbol{\theta}^T = (\theta_1, \theta_2, \ldots, \theta_n)$ can be written as a linear combination of the rows of V using multipliers $\boldsymbol{\alpha}^T \equiv (\alpha_0, \alpha_1, \ldots, \alpha_{\lceil \log_2 n \rceil})$ so that $\boldsymbol{\alpha}^T V = \boldsymbol{\theta}^T$. Then (5) simplifies to

$$x = \alpha_0 + \sum_{k=1}^{\lceil \log_2 n \rceil} \alpha_k u_k, \ \boldsymbol{u} \text{ binary}, \ (\boldsymbol{u}, \boldsymbol{\lambda}) \in \Lambda'. \tag{7}$$

Since x in (7) is described entirely in terms of $\boldsymbol{u}$, the variables $\boldsymbol{\lambda}$ simply ensure that the $\boldsymbol{u}$ vector is a column $\boldsymbol{v}_j$ corresponding to the binary expansion of some integer between 0 and $n-1$. Then (7) can be rewritten as

$$x = \alpha_0 + \sum_{k=1}^{\lceil \log_2 n \rceil} \alpha_k u_k, \ \sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n-1, \ \boldsymbol{u} \text{ binary}. \tag{8}$$

Using logic similar to that for (3), if $\lceil \log_2 n \rceil = \log_2 n$, then the inequality of (8) is unnecessary. In this case, the polytope obtained by relaxing $\boldsymbol{u}$ binary to $\mathbf{0} \leq \boldsymbol{u} \leq \mathbf{1}$ can be readily shown to have $\boldsymbol{u}$ binary at all extreme points, and thus to be locally ideal. But when $\lceil \log_2 n \rceil > \log_2 n$, the locally ideal property is not preserved in the simplification step from (7) to (8), as demonstrated in the example below. However, regardless of the value of n, for those special instances where x is integer with $\theta_j = \theta_{j-1} + 1$ for $j = 2, \ldots, n$, we have $\alpha_0 = \theta_1$ and $\alpha_k = 2^{k-1}$ for $k = 1, \ldots, \lceil \log_2 n \rceil$, reducing (8) to (3).

EXAMPLE 1. Let $x \in S \equiv \{2, 3, 5, 7, 8\}$ so that $n = 5$, $\lceil \log_2 n \rceil = 3$, and $\boldsymbol{\theta} = (2, 3, 5, 7, 8)^T$. Arranging the vectors $(1, \boldsymbol{v}_j)^T$ as the columns of V, we obtain that (5) can be written as

$$x = 2\lambda_1 + 3\lambda_2 + 5\lambda_3 + 7\lambda_4 + 8\lambda_5, \quad (\boldsymbol{u}, \boldsymbol{\lambda}) \in \Lambda'$$

where

$$\Lambda' = \left\{ (\boldsymbol{u}, \boldsymbol{\lambda}) \in \mathbb{R}^3 \times \mathbb{R}^5 : V\boldsymbol{\lambda} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix} = \begin{bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \end{bmatrix}, \ \boldsymbol{u} \text{ binary}, \ \boldsymbol{\lambda} \geq \mathbf{0} \right\}.$$

There exists no $\boldsymbol{\alpha}$ with $\boldsymbol{\alpha}^T V = \boldsymbol{\theta}^T$ and hence the $\boldsymbol{\lambda}$ variables cannot be removed. If, however, $S = \{2, 3, 5, 6, 8\}$, then $\boldsymbol{\alpha}^T V = \boldsymbol{\theta}^T$ for $\boldsymbol{\alpha}^T = (2, 1, 3, 6)$ and we can obtain (8) with

$$x = 2 + u_1 + 3u_2 + 6u_3, \ u_1 + 2u_2 + 4u_3 \leq 4, \ \boldsymbol{u} \text{ binary}.$$

For this latter case, since $\lceil \log_2 5 \rceil > \log_2 5$, the inequality is needed and the representation is not locally ideal, with $(x, u_1, u_2, u_3) = (\frac{15}{2}, 1, 1, \frac{1}{4})$ an extreme point to the relaxation that does not have $\boldsymbol{u}$ binary.

Now, observe that (5) can be extended to express any function $f(x)$ of the discrete variable x, as well as the product of x and/or any such $f(x)$ with a nonnegative variable κ, in terms of the same $\lceil \log_2 n \rceil$ binary variables $\boldsymbol{u}$. Relative to the function $f(x)$, define a variable, say y, and include the linear equation

$$y = \sum_{j=1}^{n} f(\theta_j)\lambda_j \tag{9}$$

in (5). This equation forces y to equal $f(x)$ for binary $\boldsymbol{u}$. The products $x\kappa$ and $f(x)\kappa$ for nonnegative κ rely on a modification of (4). Suppose that each restriction in Λ' (exclusive of $\boldsymbol{u}$ binary) is multiplied by the nonnegative κ to obtain the system $\Gamma(\kappa)$ below, where we use variables $\boldsymbol{\gamma}$ to denote the scaled $\boldsymbol{\lambda}$.

$$\Gamma(\kappa) \equiv \left\{ (\boldsymbol{u}, \boldsymbol{\gamma}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n : \sum_{j=1}^{n} \gamma_j = \kappa, \ \sum_{j=1}^{n} \boldsymbol{v}_j \gamma_j = \boldsymbol{u}\kappa, \ \boldsymbol{u} \text{ binary}, \ \boldsymbol{\gamma} \geq \mathbf{0} \right\} \tag{10}$$

Then, since (10) is a scaling of the equations in (4), we have for any nonnegative realization of κ that the expressions $\sum_{j=1}^{n} \theta_j \gamma_j$ and $\sum_{j=1}^{n} f(\theta_j)\gamma_j$, which are scaled versions of that found in (5) and (9) respectively, will equal the products $x\kappa$ and $y\kappa$.

A drawback of (10) is that $\lceil \log_2 n \rceil$ of the equations contain quadratic terms, as found in the vector $\boldsymbol{u}\kappa$. These terms can be linearized via a procedure of Glover (1975) that replaces $\boldsymbol{u}\kappa$ with a vector of continuous variables $\boldsymbol{w}$, and enforces $\boldsymbol{w} = \boldsymbol{u}\kappa$ using the $4\lceil \log_2 n \rceil$ inequalities below. Here κ^- and κ^+ are lower and upper bounds on the permissible values of κ, and $\mathbf{1}$ represents a vector of ones in $\mathbb{R}^{\lceil \log_2 n \rceil}$.

$$\kappa^- \boldsymbol{u} \leq \boldsymbol{w} \leq \kappa^+ \boldsymbol{u} \text{ and } \kappa\mathbf{1} - \kappa^+(\mathbf{1} - \boldsymbol{u}) \leq \boldsymbol{w} \leq \kappa\mathbf{1} - (\mathbf{1} - \boldsymbol{u})\kappa^- \tag{11}$$

For each $k \in \{1, \ldots, \lceil \log_2 n \rceil\}$, if $u_k = 0$, the left-hand inequalities enforce $w_k = 0$ and the right-hand inequalities are redundant, while if $u_k = 1$, the right-hand inequalities enforce $w_k = \kappa$ and the left-hand inequalities are redundant.

We denote the linearized version of $\Gamma(\kappa)$ where $\boldsymbol{w}$ is substituted in (10) for $\boldsymbol{u}\kappa$ using (11) by $\Gamma'(\kappa)$, as given below.

$$\Gamma'(\kappa) \equiv \left\{ \begin{array}{l} (\boldsymbol{u}, \boldsymbol{\gamma}, \boldsymbol{w}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n \times \mathbb{R}^{\lceil \log_2 n \rceil} : \\ \qquad \sum_{j=1}^n \gamma_j = \kappa,\ \sum_{j=1}^n \boldsymbol{v}_j \gamma_j = \boldsymbol{w},\ \boldsymbol{u} \text{ binary}, \boldsymbol{\gamma} \geq \mathbf{0}, \\ \qquad \kappa^- \boldsymbol{u} \leq \boldsymbol{w} \leq \kappa^+ \boldsymbol{u} \text{ and } \kappa\mathbf{1} - \kappa^+(\mathbf{1} - \boldsymbol{u}) \leq \boldsymbol{w} \leq \kappa\mathbf{1} - (\mathbf{1} - \boldsymbol{u})\kappa^- \end{array} \right\} \tag{12}$$

Concise representations of the form given by (7) that do not require any variables $\boldsymbol{\lambda}$ can also be obtained for special cases of $f(x)$, and concise representations that do not require any variables $\boldsymbol{\gamma}$ can be similarly obtained for special cases of the functions $x\kappa$ and $f(x)\kappa$. Observe that $x\kappa$ can be expressed in such a concise form if and only if x can be so represented; that is, if and only if $\boldsymbol{\theta}^T$ can be expressed as a linear combination of the rows of V. In an analogous manner, $f(x)$ and $f(x)\kappa$ can be expressed without variables $\boldsymbol{\lambda}$ and $\boldsymbol{\gamma}$ respectively if and only if the vector $\boldsymbol{f}^T = (f(\theta_1), f(\theta_2), \ldots, f(\theta_n))$ can be expressed as a linear combination of the rows of V. Of course, if it is desired to express either *both* x and $f(x)$ without variables $\boldsymbol{\lambda}$ and/or *both* $x\kappa$ and $f(x)\kappa$ without variables $\boldsymbol{\gamma}$, then *both* vectors $\boldsymbol{\theta}^T$ and $\boldsymbol{f}^T$ must be able to be expressed as linear combinations of the rows of V.

In the next section, we consider products of discrete functions indexed by the letter ℓ. It may simplify the reading to note that the parameters n and $\boldsymbol{v}_j$, the function $f(x)$, the variables x, y, κ, λ_j, γ_j, and $\boldsymbol{w}$, the binary values $\boldsymbol{u}$, and the sets S, Λ', and $\Gamma'(\kappa)$ all play the same role in that section as they do here, with an additional subscript used to indicate the associated function $f_\ell(x_\ell)$ under consideration. Where appropriate, this same notation is also used in Section 4.

3. Base-2 Representations of Products of Discrete Functions

The strategy of (4) and (10) to transform the n binary $\boldsymbol{\lambda}$ and the n binary $\boldsymbol{\gamma}$ to nonnegative continuous variables through the defining of $\lceil \log_2 n \rceil$ new binary $\boldsymbol{u}$, combined with the linearization of the expressions $\boldsymbol{u}\kappa$ of (10) via (11) to obtain (12), can be used to construct concise mixed 0-1 linear representations of products of functions of discrete variables. This construction yields representations that dominate the two methods of Li and Lu (2009) in terms of numbers of constraints, while affording improved relaxation strength relative to the first approach and equivalent strength relative to the second.

Consider m functions $f_\ell(x_\ell)$, $\ell \in \{1, \dots, m\}$, where $x_\ell \in S_\ell \equiv \{\theta_{\ell 1}, \theta_{\ell 2}, \dots, \theta_{\ell n_\ell}\}$ and where n_ℓ denotes the number of realizations of x_ℓ. Here, we subscript the function $f(x)$, the variable x, the set S, and the multiplier κ of the previous sections with the index ℓ to denote the m different functions. Also, we let $\theta_{\ell j}$ denote the j^{th} realization of the variable x_ℓ. We further construct sets Λ'_ℓ and $\Gamma'_\ell(\kappa_\ell)$ of the form (4) and (12) respectively, one corresponding to each function $f_\ell(x_\ell)$, and accordingly apply the subscript ℓ to the variables $\boldsymbol{u}$, $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$, and $\boldsymbol{w}$, as well as to the vectors $\boldsymbol{v}_j$, to obtain the sets, for each $\ell \in \{1, \dots, m\}$, given as

$$\Lambda'_\ell \equiv \left\{ (\boldsymbol{u}_\ell, \boldsymbol{\lambda}_\ell) \in \mathbb{R}^{\lceil \log_2(n_\ell) \rceil} \times \mathbb{R}^{n_\ell} : \sum_{j=1}^{n_\ell} \lambda_{\ell j} = 1,\ \sum_{j=1}^{n_\ell} \boldsymbol{v}_{\ell j} \lambda_{\ell j} = \boldsymbol{u}_\ell,\ \boldsymbol{u}_\ell \text{ binary},\ \boldsymbol{\lambda}_\ell \geq \boldsymbol{0} \right\},$$

and

$$\Gamma'_\ell(\kappa_\ell) \equiv \left\{ \begin{array}{l} (\boldsymbol{u}_\ell, \boldsymbol{\gamma}_\ell, \boldsymbol{w}_\ell) \in \mathbb{R}^{\lceil \log_2(n_\ell) \rceil} \times \mathbb{R}^{n_\ell} \times \mathbb{R}^{\lceil \log_2(n_\ell) \rceil} : \\ \qquad \sum_{j=1}^{n_\ell} \gamma_{\ell j} = \kappa_\ell,\ \sum_{j=1}^{n_\ell} \boldsymbol{v}_{\ell j} \gamma_{\ell j} = \boldsymbol{w}_\ell,\ \boldsymbol{u}_\ell \text{ binary},\ \boldsymbol{\gamma}_\ell \geq \boldsymbol{0}, \\ \qquad \kappa_\ell^- \boldsymbol{u}_\ell \leq \boldsymbol{w}_\ell \leq \kappa_\ell^+ \boldsymbol{u}_\ell \text{ and } \kappa_\ell \boldsymbol{1} - \kappa_\ell^+ (\boldsymbol{1} - \boldsymbol{u}_\ell) \leq \boldsymbol{w}_\ell \leq \kappa_\ell \boldsymbol{1} - (\boldsymbol{1} - \boldsymbol{u}_\ell)\kappa_\ell^- \end{array} \right\} \tag{13}$$

where κ_ℓ^- and κ_ℓ^+ denote lower and upper bounds on the values of κ_ℓ.

By the logic of the previous sections, for each $\ell \in \{1, \dots, m\}$, the variable x_ℓ and function $f_\ell(x_\ell)$ can be expressed as in (5) and (9) by

$$x_\ell = \sum_{j=1}^{n_\ell} \theta_{\ell j} \lambda_{\ell j} \text{ and } y_\ell = \sum_{j=1}^{n_\ell} f_\ell(\theta_{\ell j}) \lambda_{\ell j},\ (\boldsymbol{u}_\ell, \boldsymbol{\lambda}_\ell) \in \Lambda'_\ell, \tag{14}$$

where $y_\ell = f_\ell(x_\ell)$, and the products $x_\ell\kappa_\ell$ and $f_\ell(x_\ell)\kappa_\ell$ can be expressed by

$$x_\ell\kappa_\ell = \sum_{j=1}^{n_\ell}\theta_{\ell j}\gamma_{\ell j} \text{ and } f_\ell(x_\ell)\kappa_\ell = \sum_{j=1}^{n_\ell} f_\ell(\theta_{\ell j})\gamma_{\ell j},\ (\boldsymbol{u}_\ell,\boldsymbol{\gamma}_\ell,\boldsymbol{w}_\ell)\in\Gamma'_\ell(\kappa_\ell). \tag{15}$$

If desired, the products $x_\ell\kappa_\ell$ and $f_\ell(x_\ell)\kappa_\ell$ can each be replaced in (15) by continuous variables.

We now focus on a representation of the product $\prod_{j=1}^m f_j(x_j)$ using the sets Λ'_ℓ and $\Gamma'_\ell(\kappa_\ell)$ from above. To begin, for each $\ell\in\{2,\dots,m\}$, we represent the product $f_1(x_1)f_2(x_2)$ by a continuous variable y_{12}, the product $f_1(x_1)f_2(x_2)f_3(x_3)$ by a variable y_{123}, and so on up to the product $f_1(x_1)f_2(x_2)\cdots f_m(x_m)$ by a variable $y_{12\cdots m}$. For ease of notation, for each $\ell\in\{1,\dots,m\}$, let $J_\ell = 1\cdots\ell$ denote consecutive subscript indices so that $\prod_{j=1}^\ell f_j(x_j)$ is represented by the variable y_{J_ℓ} (with $y_1 = y_{J_1}$). As additional notation, for each $\ell\in\{1,\dots,m-1\}$, denote computed lower and upper bounds on the product $\prod_{j=1}^\ell f_j(x_j)$ by $f^-_{J_\ell}$ and $f^+_{J_\ell}$ respectively. Continue by constructing Λ'_ℓ and expressing the variables x_ℓ and y_ℓ as in (14) for each $\ell\in\{1,\dots,m\}$. Then compute $\Gamma'_\ell(\kappa_\ell)$ of (13) for each $\ell\in\{2,\dots,m\}$ with the nonnegative scalar κ_ℓ given by $\kappa_\ell = \prod_{j=1}^{\ell-1} f_j(x_j) - f^-_{J_{\ell-1}}$. Such κ_ℓ have lower and upper bounds of $\kappa^-_\ell = 0$ and $\kappa^+_\ell = f^+_{J_{\ell-1}} - f^-_{J_{\ell-1}}$ respectively. The resulting system follows where, for each $\ell\in\{2,\dots,m\}$, we have included explicit restrictions that $\kappa_\ell = y_{J_{\ell-1}} - f^-_{J_{\ell-1}}$, with $y_{J_{\ell-1}}$ substituted for the linearized version of $\prod_{j=1}^{\ell-1} f_j(x_j)$.

$$x_\ell = \sum_{j=1}^{n_\ell}\theta_{\ell j}\lambda_{\ell j},\ y_\ell = \sum_{j=1}^{n_\ell} f_\ell(\theta_{\ell j})\lambda_{\ell j},\ (\boldsymbol{u}_\ell,\boldsymbol{\lambda}_\ell)\in\Lambda'_\ell\ \ \forall\,\ell = 1,\dots,m \tag{16}$$

$$\kappa_\ell = y_{J_{\ell-1}} - f^-_{J_{\ell-1}}\ \ \forall\,\ell = 2,\dots,m \tag{17}$$

$$y_{J_\ell} = \sum_{j=1}^{n_\ell} f_\ell(\theta_{\ell j})\gamma_{\ell j} + y_\ell f^-_{J_{\ell-1}},\ (\boldsymbol{u}_\ell,\boldsymbol{\gamma}_\ell,\boldsymbol{w}_\ell)\in\Gamma'_\ell(\kappa_\ell)\ \ \forall\,\ell = 2,\dots,m \tag{18}$$

Note that the $\boldsymbol{u}_\ell$ binary restrictions for $\ell\in\{2,\dots,m\}$ are found in both (16) and (18) but need only be stated once. The desired result that $y_{J_i} = \prod_{j=1}^i f_j(x_j)$ for $i = 1,\dots,m$ can be envisioned as inductively obtained. The base case having $i=1$ is established by (16) with $\ell = 1$ to yield $y_1 = y_{J_1} = f_1(x_1)$. For each $i\in\{2,\dots,m\}$, the argument assumes that $y_{J_{i-1}} = \prod_{j=1}^{i-1} f_j(x_j)$, and then uses restrictions (16)–(18) with $\ell = i$ to enforce that $y_{J_i} = f_i(x_i)y_{J_{i-1}}$. The products $x_\ell\kappa_\ell$ of (15) do not appear in (16)–(18) as they are not needed in the representation of $\prod_{j=1}^m f_j(x_j)$.

Upon substituting $\kappa_\ell = y_{J_{\ell-1}} - f^-_{J_{\ell-1}}$ for each $\ell \in \{2,\dots,m\}$ from (17) into (18) and then removing (17), the counts on the types and numbers of variables in (16) and (18) are summarized in Table 1. Summing relevant entries, Table 1 gives that (16) and (18) have a total of $3m-1+n_1+2\sum_{\ell=2}^m n_\ell + \sum_{\ell=2}^m \lceil \log_2(n_\ell) \rceil$ continuous variables and $\sum_{\ell=1}^m \lceil \log_2(n_\ell) \rceil$ binary variables.

Table 1 Variable types and counts in (16) and (18).

Variable name	Variable type	Number of such variables
x_ℓ	continuous	m
y_ℓ	continuous	m
$y_{J_\ell}, \ell \neq 1$	continuous	$m-1$
λ_ℓ	continuous	n_ℓ for each $\ell \in \{1,\dots,m\}$
$\boldsymbol{\gamma}_\ell$	continuous	n_ℓ for each $\ell \in \{2,\dots,m\}$
$\boldsymbol{w}_\ell$	continuous	$\lceil \log_2(n_\ell) \rceil$ for each $\ell \in \{2,\dots,m\}$
$\boldsymbol{u}_\ell$	binary	$\lceil \log_2(n_\ell) \rceil$ for each $\ell \in \{1,\dots,m\}$

Relative to the number of constraints in (16) and (18), a count is as follows. Each set Λ'_ℓ of (16) has $\lceil \log_2(n_\ell) \rceil + 1$ restrictions, while each set $\Gamma'_\ell(\kappa_\ell)$ of (18) with κ_ℓ as defined in (17) has $5\lceil \log_2(n_\ell) \rceil + 1$ restrictions. Including the additional $2m$ equalities defining x_ℓ and y_ℓ of (16) and the $m-1$ equalities defining y_{J_ℓ} for $\ell \neq 1$ of (18), the total number of constraints is $5m-2+\lceil \log_2(n_1) \rceil + 6\sum_{\ell=2}^m \lceil \log_2(n_\ell) \rceil$.

The numbers of variables and constraints can be reduced, depending on the structure of the problem and the desired form of the resulting linearization. Four reduction strategies are listed below.

1. Since $\kappa^-_\ell = 0$ for each $\ell \in \{2,\dots,m\}$, the inequalities $\kappa^-_\ell \boldsymbol{u}_\ell \leq \boldsymbol{w}_\ell$ of (13) become nonnegativity on $\boldsymbol{w}_\ell$, reducing the number of constraints by $\sum_{\ell=2}^m \lceil \log_2(n_\ell) \rceil$. If some κ_ℓ is defined which allows for a strengthening of κ^-_ℓ from 0 to a positive value, then a transformation of variables $\boldsymbol{w}'_\ell = \boldsymbol{w}_\ell - \kappa^-_\ell \boldsymbol{u}_\ell$ as in Adams and Forrester (2005) and Glover (1984) can be used.

2. If desired, the variables x_ℓ, y_ℓ, and y_{J_ℓ} can all be substituted from the linearization (as well as any encompassing optimization problem) by using the definition of variables in terms of $\lambda_{\ell j}$ and $\gamma_{\ell j}$ found in (16) and (18). This substitution reduces the number of variables and constraints by $3m-1$ each.

3. Each of the sets Λ'_ℓ and $\Gamma'_\ell(\kappa_\ell)$ can be reduced in size by $\lceil \log_2(n_\ell) \rceil + 1$ variables via a transformation that changes the equality restrictions to inequality. To see this, consider Λ'_1. As the defining linear system of equations is of full rank (choose the columns corresponding to λ_{11} and $\lambda_{1(2^{p-1}+1)}$ for each $p \in \{1, \ldots, \lceil \log_2(n_1) \rceil\}$), a basis for $\mathbb{R}^{\lceil \log_2(n_1) \rceil + 1}$ can be obtained in terms of a subset of the columns of the defining system. Then the $\lceil \log_2(n_1) \rceil + 1$ basic variables can be expressed in terms of the nonbasic variables and subsequently eliminated from the formulation. Performing such a reduction on each Λ'_ℓ and $\Gamma'_\ell(\kappa_\ell)$ reduces the formulation by $2m - 1 + \lceil \log_2(n_1) \rceil + 2\sum_{\ell=2}^{m} \lceil \log_2(n_\ell) \rceil$ continuous variables.

4. The order in which the functions are numbered and subsequently linearized affects the variable and constraint counts. The set $\Gamma'_1(\kappa_1)$ of (13) does not appear in (18), nor do the associated variables $\boldsymbol{\gamma}_1$ and $\boldsymbol{w}_1$. Therefore, selecting $f_1(x_1)$ so that $n_1 = \max\{n_\ell : \ell = 1, \ldots, m\}$ can yield a smaller formulation.

The lower and upper bounds $f^-_{J_\ell}$ and $f^+_{J_\ell}$ on the products $\prod_{j=1}^{\ell} f_j(x_j)$ for $\ell \in \{1, \ldots, m-1\}$ used in the construction of (16)–(18) can be computed in different ways. For each $\ell \in \{1, \ldots, m\}$, lower and upper bounds f^-_ℓ and f^+_ℓ on the function $f_\ell(x_\ell)$ are readily obtained as $f^-_\ell = \min\{f_\ell(\theta_{\ell j}) : j = 1, \ldots, n_\ell\}$ and $f^+_\ell = \max\{f_\ell(\theta_{\ell j}) : j = 1, \ldots, n_\ell\}$. Next consider the values $f^-_{J_\ell}$ and $f^+_{J_\ell}$ for $\ell \in \{2, \ldots, m-1\}$. If $f^-_j \geq 0$ for all $j \in \{1, \ldots, \ell\}$, then we can use $f^-_{J_\ell} = \prod_{j=1}^{\ell} f^-_j$ and $f^+_{J_\ell} = \prod_{j=1}^{\ell} f^+_j$. If, however, $f^-_j < 0$ for some such j, then various options exist, including using $f^+_{J_\ell} = \prod_{j=1}^{\ell} \max\{|f^-_j|, |f^+_j|\}$ and $f^-_{J_\ell} = -f^+_{J_\ell}$.

Three additional remarks relative to (16)–(18) are warranted. First, products of discrete variables (as opposed to products of functions of discrete variables) can be readily handled by having $f_\ell(x_\ell)$ serve as identity functions so that $f_\ell(\theta_{\ell j}) = \theta_{\ell j}$ for all $\ell \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n_\ell\}$. Then the first equation in (16) defining x_ℓ can be removed for each $\ell \in \{1, \ldots, m\}$, as $x_\ell = y_\ell$. Second, the linearization process that produces (16)–(18) does not depend on x_1 being discrete. This allows us to accommodate the expression $\prod_{j=1}^{m} f_j(x_j)$ when the function $f_1(x_1)$ is continuous. In this case, restrictions (16) with $\ell = 1$ are not used. Third, the approach of (16)–(18) does not make use of the product $f_1(x_1)\kappa_1$, so the value κ_1 and set $\Gamma'_1(\kappa_1)$ of (13) is not found in (18). Similarly, the lower and upper bounds $f^-_{J_m}$ and $f^+_{J_m}$ on $\prod_{j=1}^m f_j(x_j)$ are not needed.

We conclude this section with an example demonstrating the use of (16) and (18) in linearizing the monomial $x_1^3 x_2^{1.5}$.

EXAMPLE 2. Consider the $m=2$ functions $f_1(x_1)=x_1^3$ and $f_2(x_2)=x_2^{1.5}$, where $x_1 \in S_1 \equiv \{-1,2,5,7\}$ and $x_2 \in S_2 \equiv \{2,4,8\}$, so that $n_1=4$ and $n_2=3$. The restrictions (16) and (18) have the continuous variables y_1, y_2, and y_{12} replacing $f_1(x_1)$, $f_2(x_2)$, and the product $f_1(x_1)f_2(x_2)=x_1^3x_2^{1.5}$ respectively. Using matrices to simplify notation where possible, (16) is given by

$$x_1 = -1\lambda_{11} + 2\lambda_{12} + 5\lambda_{13} + 7\lambda_{14},\;\; y_1 = (-1)^3\lambda_{11} + 2^3\lambda_{12} + 5^3\lambda_{13} + 7^3\lambda_{14},\;\; (\boldsymbol{u}_1, \boldsymbol{\lambda}_1) \in \Lambda'_1,$$

where

$$\Lambda'_1 = \left\{ (\boldsymbol{u}_1, \boldsymbol{\lambda}_1) \in \mathbb{R}^2 \times \mathbb{R}^4 : \begin{bmatrix} 1&1&1&1 \\ 0&1&0&1 \\ 0&0&1&1 \end{bmatrix} \begin{bmatrix} \lambda_{11} \\ \lambda_{12} \\ \lambda_{13} \\ \lambda_{14} \end{bmatrix} = \begin{bmatrix} 1 \\ u_{11} \\ u_{12} \end{bmatrix}, \boldsymbol{u}_1 \text{ binary}, \boldsymbol{\lambda}_1 \geq \mathbf{0} \right\},$$

and

$$x_2 = 2\lambda_{21} + 4\lambda_{22} + 8\lambda_{23},\;\; y_2 = 2^{1.5}\lambda_{21} + 4^{1.5}\lambda_{22} + 8^{1.5}\lambda_{23},\;\; (\boldsymbol{u}_2, \boldsymbol{\lambda}_2) \in \Lambda'_2,$$

where

$$\Lambda'_2 = \left\{ (\boldsymbol{u}_2, \boldsymbol{\lambda}_2) \in \mathbb{R}^2 \times \mathbb{R}^3 : \begin{bmatrix} 1&1&1 \\ 0&1&0 \\ 0&0&1 \end{bmatrix} \begin{bmatrix} \lambda_{21} \\ \lambda_{22} \\ \lambda_{23} \end{bmatrix} = \begin{bmatrix} 1 \\ u_{21} \\ u_{22} \end{bmatrix}, \boldsymbol{u}_2 \text{ binary}, \boldsymbol{\lambda}_2 \geq \mathbf{0} \right\}.$$

Since $f^-_{J_1} = f^-_1 = (-1)^3$, we have $\kappa_2 = x_1^3 - (-1)^3 = y_1 + 1$, with $\kappa^-_2 = 0$ and $\kappa^+_2 = f^+_1 - f^-_1 = 7^3 - (-1)^3 = 344$. Then (18) becomes

$$y_{12} = 2^{1.5}\gamma_{21} + 4^{1.5}\gamma_{22} + 8^{1.5}\gamma_{23} - y_2,\;\; (\boldsymbol{u}_2, \boldsymbol{\gamma}_2, \boldsymbol{w}_2) \in \Gamma'_2(y_1+1),$$

where $\Gamma'(y_1+1)$ of (13) is expressed in matrix form as

$$\Gamma'_2(y_1+1) = \left\{ \begin{array}{c} (\boldsymbol{u}_2, \boldsymbol{\gamma}_2, \boldsymbol{w}_2) \in \mathbb{R}^2 \times \mathbb{R}^3 \times \mathbb{R}^2, \boldsymbol{\gamma}_2 \geq \mathbf{0} : \\ \begin{bmatrix} 1&1&1 \\ 0&1&0 \\ 0&0&1 \end{bmatrix} \begin{bmatrix} \gamma_{21} \\ \gamma_{22} \\ \gamma_{23} \end{bmatrix} = \begin{bmatrix} y_1+1 \\ w_{21} \\ w_{22} \end{bmatrix}, \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix} \leq 344 \begin{bmatrix} u_{21} \\ u_{22} \end{bmatrix}, \\ \begin{bmatrix} y_1+1 \\ y_1+1 \end{bmatrix} - 344 \begin{bmatrix} 1-u_{21} \\ 1-u_{22} \end{bmatrix} \leq \begin{bmatrix} w_{21} \\ w_{22} \end{bmatrix} \leq \begin{bmatrix} y_1+1 \\ y_1+1 \end{bmatrix} \end{array} \right\},$$

with $\boldsymbol{u}_2$ binary not explicitly listed as it is found in Λ'_2 above. Note that while the sets Λ'_1 and Λ'_2 were earlier explained to be locally ideal, the presence of $\Gamma'_2(y_1+1)$ forfeits this property. Upon removing the binary restrictions on $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, a nonintegral extreme point is given by $\boldsymbol{u}_1 = \left(\frac{1}{3}, 0\right)$, $\boldsymbol{u}_2 = (0,1)$, $\boldsymbol{\lambda}_1 = \left(\frac{2}{3}, \frac{1}{3}, 0, 0\right)$, $\boldsymbol{\lambda}_2 = (0,0,1)$, $\boldsymbol{w}_2 = (0,3)$, and $\boldsymbol{\gamma}_2 = (0,0,3)$, with $(x_1, x_2, y_1, y_2, y_{12}) = (0, 8, 2, 16\sqrt{2}, 32\sqrt{2})$. Now, suppose that we change the problem so that the variable x_1 is redefined to be *continuous* in the interval $[-1, 7]$, and it is desired to have y_{12} represent the product of the *continuous* function x_1^3 having $-1 \le x_1 \le 7$ with the discrete-valued function $x_2^{1.5}$ having $x_2 \in S_2$, so that again $y_{12} = x_1^3 x_2^{1.5}$. Explicitly define y_1 to be x_1^3 via $y_1 = x_1^3$, and treat y_1 as a continuous function with $y_1 \in [-1, 343]$. In this case, none of the restrictions associated with (16) having $\ell = 1$ are needed (including Λ_1') and the values $f_1^- = -1$, $f_1^+ = 343$, $\kappa_2^- = 0$, and $\kappa_2^+ = 344$ are unchanged so that the set $\Gamma_2'(\kappa_2) = \Gamma_2'(y_1 + 1)$ remains the same.

4. Comparisons with Other Methods

The size and relaxation strength of the system (16)–(18) compares favorably with alternate approaches. While there is considerable literature dealing with the linearization of nonlinear 0-1 programs and the representation of discrete variables in terms of binary variables, little attention has been given to modeling functions of discrete variables, and their products, in terms of logarithmic numbers of binary variables. We focus attention here on the two methods from Li and Lu Li and Lu (2009), one per each of the first two subsections below. These methods were reportedly designed for solving mixed-discrete generalized geometric programs.

4.1. Li & Lu Approach 1

Given a discrete variable x that can realize values in the set $S = \{\theta_1, \theta_2, \ldots, \theta_n\}$ and a function $f(x)$ defined in terms of x, the first approach of Li and Lu (2009) linearizes $f(x)$ using $\lceil \log_2 n \rceil$ binary variables and $2n+1$ linear inequalities, plus a single continuous variable y to represent $f(x)$. We temporarily adopt the notation of Section 2 that suppresses the subscript ℓ on the variable x, the function $f(x)$, the set S, the parameter n, the values θ_j for $j \in \{1, \ldots, n\}$, and the vectors $\boldsymbol{u}$, $\boldsymbol{\lambda}$, and $\boldsymbol{v}_j$ since a single function of a discrete variable is initially considered.

This approach of Li and Lu (2009) can be explained in terms of ours as follows. It uses the same binary variables $\boldsymbol{u} \in \mathbb{R}^{\lceil \log_2 n \rceil}$ as (4) with (9), but in an altogether different manner. While not defining vectors $\boldsymbol{v}_j$ or variables $\boldsymbol{\lambda}$, it can be envisioned as also enforcing that $y = f(\theta_j)$ when $\boldsymbol{u} = \boldsymbol{v}_j$. (For now, we focus attention on the function $f(x)$ and later explain how the discrete variable x can be similarly handled. This method is unique in that it requires separate families of restrictions to handle each of x and $f(x)$.) For every $j \in \{1,\dots,n\}$, it defines a linear function $A_j(\boldsymbol{u})$ of the binary variables $\boldsymbol{u}$ so that $A_j(\boldsymbol{u}) = 0$ if $\boldsymbol{u} = \boldsymbol{v}_j$ and $A_j(\boldsymbol{u}) \geq 1$ if $\boldsymbol{u} \neq \boldsymbol{v}_j$. For each such j, this is accomplished by adding to the sum $\sum_{k=1}^{\lceil \log_2 n \rceil} u_k$, the expression $1 - 2u_i$ for all i having the i^{th} component of $\boldsymbol{v}_j$ as 1. These functions can be computed using matrix multiplication as follows. Define the $(\lceil \log_2 n \rceil + 1) \times (\lceil \log_2 n \rceil + 1)$ invertible, symmetric matrix B whose $(i,j)^{th}$ element, denoted B_{ij} for all $i,j \in \{1,\dots,\lceil \log_2 n \rceil + 1\}$, is given by

$$B_{ij} = \begin{cases} 1 & \text{if } (i=1 \text{ and } j \neq 1) \text{ or } (i \neq 1 \text{ and } j = 1) \\ -2 & \text{if } i = j \neq 1 \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

so that

$$\begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}^T B \begin{bmatrix} 1 \\ \boldsymbol{v}_j \end{bmatrix} = A_j(\boldsymbol{u}) = \begin{bmatrix} 1 \\ \boldsymbol{v}_j \end{bmatrix}^T B \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix} \;\; \forall\, j \in \{1,\dots,n\}. \tag{20}$$

The left equation becomes clear upon observing that the row vector $\begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}^T B \in \mathbb{R}^{\lceil \log_2 n \rceil + 1}$ has its first entry as $\sum_{k=1}^{\lceil \log_2 n \rceil} u_k$, and its i^{th} entry as $1 - 2u_{i-1}$ for each $i \in \{2,\dots,\lceil \log_2 n \rceil + 1\}$. The equality of the right expression with the left follows from $\begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}^T B \begin{bmatrix} 1 \\ \boldsymbol{v}_j \end{bmatrix}$ being a 1×1 matrix, with B symmetric. Letting $M = f^+ - f^-$ with $f^- \equiv \min\{f(\theta_1),\dots,f(\theta_n)\}$ and $f^+ \equiv \max\{f(\theta_1),\dots,f(\theta_n)\}$, this formulation of Li and Lu (2009) is as follows.

$$P \equiv \left\{ \begin{array}{l} (\boldsymbol{u}, y) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R} : \\ \qquad f(\theta_j) - M A_j(\boldsymbol{u}) \leq y \leq f(\theta_j) + M A_j(\boldsymbol{u}) \;\; \forall\, j \in \{1,\dots,n\}, \\ \qquad \displaystyle\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n-1, \\ \qquad \boldsymbol{u} \text{ binary} \end{array} \right\}$$

The restrictions of P operate so that, given any binary $\boldsymbol{u}$ satisfying $\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n-1$, the single $A_j(\boldsymbol{u})$ equaling 0, say $A_p(\boldsymbol{u})$, will have the two inequalities $f(\theta_p) - M A_p(\boldsymbol{u}) \leq f(x) \leq f(\theta_p) + M A_p(\boldsymbol{u})$ enforcing $y = f(\theta_p)$, and the remaining $2(n-1)$ inequalities with $A_j(\boldsymbol{u}) \geq 1$ being redundant.

Observe that P contains no variables $\boldsymbol{\lambda}$; it has a single continuous y and $\lceil \log_2 n \rceil$ binary $\boldsymbol{u}$. However, it requires $2n+1$ inequalities. In contrast, Λ' of (4) has n continuous $\boldsymbol{\lambda}$ and $\lceil \log_2 n \rceil$ binary $\boldsymbol{u}$, but only $\lceil \log_2 n \rceil + 1$ constraints. Recall, though, that reduction strategy 3 of Section 3 allows us to reduce the number of variables $\boldsymbol{\lambda}$ in Λ' by $\lceil \log_2 n \rceil + 1$. Thus, in summary, Λ' and P require the same number of binary variables, but the former uses $2n - \lceil \log_2 n \rceil$ fewer constraints at the expense of $n - \lceil \log_2 n \rceil - 2$ more continuous variables.

An important consideration when expressing any function of a discrete variable in terms of binary variables in a mixed 0-1 linear form is the strength of the continuous relaxation. Let $\bar{\Lambda}'$ and $\bar{P}$ denote, respectively, the continuous relaxations of Λ' and P obtained by relaxing the $\boldsymbol{u}$ binary restrictions to $\mathbf{0} \le \boldsymbol{u} \le \mathbf{1}$. (Note that these $2\lceil \log_2 n \rceil$ inequalities are not needed in the set $\bar{\Lambda}'$, as they are implied by the other restrictions.) The theorem below shows that the set $\bar{\Lambda}'$ with (9) provides at least as tight a polyhedral representation, in terms of permissible values of y, as does $\bar{P}$.

THEOREM 1. *Given any* $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}}) \in \bar{\Lambda}'$ *of (4), we have* $(\hat{\boldsymbol{u}}, \hat{y}) \in \bar{P}$, *where* $\hat{y} = \sum_{j=1}^n f(\theta_j)\hat{\lambda}_j$.

Proof of Theorem 1. Let $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}}) \in \bar{\Lambda}'$ with $\hat{y} = \sum_{j=1}^n f(\theta_j)\hat{\lambda}_j$. Since for each $j \in \{1, \ldots, n\}$, $\boldsymbol{u} = \boldsymbol{v}_j$ satisfies $\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \le n-1$, and since $\bar{\Lambda}'$ expresses $\hat{\boldsymbol{u}}$ as a convex combination $\hat{\boldsymbol{\lambda}}$ of the vectors $\boldsymbol{v}_j$, it follows that $\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} \hat{u}_k \le n-1$. Thus, the proof reduces to showing that

$$f(\theta_j) - MA_j(\hat{\boldsymbol{u}}) \le \hat{y} \le f(\theta_j) + MA_j(\hat{\boldsymbol{u}}) \; \forall \, j \in \{1, \ldots, n\}. \tag{21}$$

Toward this end, arbitrarily select any $p \in \{1, \ldots, n\}$ and consider (21) for $j = p$. Surrogate the equations of $\bar{\Lambda}'$, represented in matrix form as in (6), using the multipliers $\begin{bmatrix} 1 \\ \boldsymbol{v}_p \end{bmatrix}^T B$, and set $(\boldsymbol{u}, \boldsymbol{\lambda}) = (\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}})$, to obtain

$$\sum_{\substack{j=1 \\ j \ne p}}^n \hat{\lambda}_j \le \sum_{j=1}^n A_j(\boldsymbol{v}_p)\hat{\lambda}_j = \begin{bmatrix} 1 \\ \boldsymbol{v}_p \end{bmatrix}^T BV\hat{\boldsymbol{\lambda}} = \begin{bmatrix} 1 \\ \boldsymbol{v}_p \end{bmatrix}^T B \begin{bmatrix} 1 \\ \hat{\boldsymbol{u}} \end{bmatrix} = A_p(\hat{\boldsymbol{u}}). \tag{22}$$

The inequality follows from the nonnegativity of $\hat{\boldsymbol{\lambda}}$ and because the function $A_j(\boldsymbol{v}_p)$ is defined to have $A_p(\boldsymbol{v}_p) = 0$ and $A_j(\boldsymbol{v}_p) \ge 1$ for $j \ne p$. The first equality is due to the left equation of (20) with $\boldsymbol{u} = \boldsymbol{v}_p$, applied once for each $j \in \{1, \ldots, n\}$. The middle equality is the surrogation of the restrictions in $\bar{\Lambda}'$, and the last equality follows from the right equation of (20) with $j = p$. Now, add the nonnegative multiple $(f^+ - f(\theta_p))$ of the inequality $\sum_{j=1, j\neq p}^{n} \hat{\lambda}_j \leq A_p(\hat{\boldsymbol{u}})$ of (22) to the multiple $f(\theta_p)$ of the equation $\sum_{j=1}^{n} \hat{\lambda}_j = 1$ from (4) to obtain

$$\sum_{j=1}^{n} f(\theta_j)\hat{\lambda}_j + \sum_{\substack{j=1 \\ j\neq p}}^{n} (f^+ - f(\theta_j))\hat{\lambda}_j \leq f(\theta_p) + (f^+ - f(\theta_p))A_p(\hat{\boldsymbol{u}})$$

which, by the nonnegativity of $(f^+ - f(\theta_j))\hat{\lambda}_j$ for all $j \neq p$ and the defining of $\hat{y} = \sum_{j=1}^{n} f(\theta_j)\hat{\lambda}_j$, establishes the right-hand inequality of (21) for $j = p$ because $f^+ - f(\theta_p) \leq f^+ - f^- = M$. Similarly, add the nonpositive multiple $(f^- - f(\theta_p))$ of the inequality $\sum_{j=1, j\neq p}^{n} \hat{\lambda}_j \leq A_p(\hat{\boldsymbol{u}})$ of (22) to the multiple $f(\theta_p)$ of the equation $\sum_{j=1}^{n} \hat{\lambda}_j = 1$ from (4) to obtain

$$\sum_{j=1}^{n} f(\theta_j)\hat{\lambda}_j + \sum_{\substack{j=1 \\ j\neq p}}^{n} (f^- - f(\theta_j))\hat{\lambda}_j \geq f(\theta_p) + (f^- - f(\theta_p))A_p(\hat{\boldsymbol{u}})$$

which, by the nonpositivity of $(f^- - f(\theta_j))\hat{\lambda}_j$ for all $j \neq p$ and the defining of $\hat{y} = \sum_{j=1}^{n} f(\theta_j)\hat{\lambda}_j$, establishes the left inequality of (21) for $j = p$ since $f^- - f(\theta_p) \geq f^- - f^+ = -M$. This completes the proof. $\square$

Note that the proof of Theorem 1 suggests a strengthening of the bound M used within P and $\bar{P}$. For each $j \in \{1, \ldots, n\}$, we can use $\underline{M}_j = f(\theta_j) - f^-$ and $\overline{M}_j = f^+ - f(\theta_j)$ to redefine the set P as

$$P \equiv \left\{ \begin{array}{l} (\boldsymbol{u}, y) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}: \\ \qquad f(\theta_j) - \underline{M}_j A_j(\boldsymbol{u}) \leq y \leq f(\theta_j) + \overline{M}_j A_j(\boldsymbol{u}) \;\; \forall\, j \in \{1, \ldots, n\}, \\ \qquad \sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n - 1, \\ \qquad \boldsymbol{u} \text{ binary} \end{array} \right\}. \tag{23}$$

The set P remains unchanged with this adjustment but $\bar{P}$ is potentially tightened.

The representation of a discrete variable x, as opposed to a function $f(x)$, proceeds in an identical manner to the above. This is readily seen by defining $f(x)$ so that $f(x) = x$. The set P of (23) will then replace each $f(\theta_j)$ with θ_j, and each occurrence of y with x. If it is desired to represent *both* $f(x)$ and x, then $4n + 1$ associated inequalities are needed in the $\lceil \log_2 n \rceil$ binary variables $\boldsymbol{u}$, as the equation $\sum_{k=1}^{\lceil \log_2 n \rceil} u_k \leq n - 1$ need not be repeated.

It is important to note that the converse of Theorem 1 is not true, even when the set $\bar{P}$ uses the improved values $\underline{M}_j$ and $\overline{M}_j$ as in (23). That is to say, there can exist a point $(\hat{\boldsymbol{u}}, \hat{y}) \in \bar{P}$ for which there exists no $\hat{\boldsymbol{\lambda}}$ having $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}}) \in \bar{\Lambda}'$ and $\hat{y} = \sum_{j=1}^{n} f(\theta_j)\hat{\lambda}_j$. An example illustrating Theorem 1 and the failure of its converse is below. For simplicity of presentation, we have $y = f(x) = x$ so that only one family of restrictions is required.

EXAMPLE 3. Consider $f(x) = x$ with $x \in S \equiv \{1, 3, 5\}$ so that $n = 3$, $\lceil \log_2 n \rceil = 2$, $f^- = 1$, and $f^+ = 5$. Then (9) with the relaxed set $\bar{\Lambda}'$ is given by

$$y = \lambda_1 + 3\lambda_2 + 5\lambda_3, \quad (\boldsymbol{u}, \boldsymbol{\lambda}) \in \bar{\Lambda}',$$

where

$$\bar{\Lambda}' = \left\{ (\boldsymbol{u}, \boldsymbol{\lambda}) \in \mathbb{R}^2 \times \mathbb{R}^3 : V\boldsymbol{\lambda} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 1 \\ u_1 \\ u_2 \end{bmatrix}, \boldsymbol{\lambda} \geq \boldsymbol{0} \right\}.$$

The set $\bar{P}$, adjusted for the strengthened $\underline{M}_j$ and $\overline{M}_j$ as in (23), is

$$\bar{P} = \left\{ \begin{array}{rcl} (\boldsymbol{u}, y) \in \mathbb{R}^2 \times \mathbb{R} : & & \\ 1 & \leq y \leq & 1 + 4(u_1 + u_2) \\ 3 - 2(1 - u_1 + u_2) & \leq y \leq & 3 + 2(1 - u_1 + u_2) \\ 5 - 4(1 + u_1 - u_2) & \leq y \leq & 5 \\ u_1 + 2u_2 & \leq 2 & \\ 0 \leq u_1 \leq 1 & & \\ 0 \leq u_2 \leq 1 & & \end{array} \right\}.$$

For $\hat{\boldsymbol{u}}^T = (\hat{u}_1, \hat{u}_2) = (1, \frac{1}{2})$, every $\hat{y}$ satisfying $\hat{y} \in [2, 4]$ will have $(\hat{\boldsymbol{u}}, \hat{y}) \in \bar{P}$. However, there exists no $\boldsymbol{\lambda}$ with $(\hat{\boldsymbol{u}}, \boldsymbol{\lambda}) \in \bar{\Lambda}'$ since the restrictions of $\bar{\Lambda}'$ enforce that the nonnegative $\boldsymbol{\lambda}$ must have $\lambda_2 = \hat{u}_1 = 1$, $\lambda_3 = \hat{u}_2 = \frac{1}{2}$, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

The paper of Li and Lu (2009) extends this approach to products of univariate functions. Again consider the m functions $f_\ell(x_\ell)$, $\ell \in \{1, \ldots, m\}$, where $x_\ell \in S_\ell \equiv \{\theta_{\ell 1}, \theta_{\ell 2}, \ldots, \theta_{\ell n_\ell}\}$ and n_ℓ denotes the number of realizations of x_ℓ. Then the linearization of $\prod_{\ell=1}^{m} f_\ell(x_\ell)$ using our strengthened bounds of (23) is accomplished in two steps. First, for each $\ell \in \{1, \ldots, m\}$, form the set P_ℓ in the same manner as (23) to represent $f_\ell(x_\ell)$ as the variable y_ℓ using the binary variables $\boldsymbol{u}_\ell \in \mathbb{R}^{\lceil \log_2(n_\ell) \rceil}$. Here, for each such ℓ and for every $j \in \{1, \ldots, n_\ell\}$, the linear functions $A_{\ell j}(\boldsymbol{u}_\ell)$ are defined in the same manner as $A_j(\boldsymbol{u})$, and the bounds $\underline{M}_{\ell j}$ and $\overline{M}_{\ell j}$ replace $\underline{M}_j$ and $\overline{M}_j$ respectively so that $\underline{M}_{\ell j} = f_\ell(\theta_{\ell j}) - f_\ell^-$

and $\overline{M}_{\ell j} = f_\ell^+ - f_\ell(\theta_{\ell j})$, with $f_\ell^- \equiv \min\{f_\ell(\theta_{\ell 1}), \ldots, f_\ell(\theta_{\ell n_\ell})\}$ and $f_\ell^+ \equiv \max\{f_\ell(\theta_{\ell 1}), \ldots, f_\ell(\theta_{\ell n_\ell})\}$. In addition, each P_ℓ has the restriction $\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_{\ell k} \le n_\ell - 1$.

The second step is based on the following observation: for any given ℓ, by multiplying the functional values $f_\ell(\theta_{\ell j})$ found within P_ℓ by a variable, say ζ, the $2n_\ell$ inequalities involving $f_\ell(\theta_{\ell j})$ will enforce $y_\ell = \zeta f_\ell(x_\ell)$ provided that for each $j \in \{1, \ldots, n_\ell\}$, the values $\underline{M}_{\ell j}$ and $\overline{M}_{\ell j}$ are adjusted so that the associated inequalities are redundant when $A_{\ell j}(\boldsymbol{u}_\ell) \ge 1$; it is sufficient to have $\zeta f_\ell(\theta_{\ell j}) - \zeta f_\ell(x_\ell) \le \underline{M}_{\ell j}$ and $\zeta f_\ell(x_\ell) - \zeta f_\ell(\theta_{\ell j}) \le \overline{M}_{\ell j}$ for all possible realizations of ζ and $f_\ell(x_\ell)$. Now, using this observation and the notation from Section 3 that $J_\ell = 1 \cdots \ell$, we can inductively have $y_{J_\ell} = \prod_{j=1}^{\ell} f_j(x_j)$ for $\ell \ge 2$, beginning with $y_{12} = f_1(x_1) f_2(x_2) = y_1 f_2(x_2)$ and sequentially progressing to $y_{J_m} = f_1(x_1) f_2(x_2) \cdots f_m(x_m) = y_{J_{m-1}} f_m(x_m)$. The variable y_{12} is computed by forming a new set P_{12} using $\zeta = y_1$ within P_2 to obtain $y_{12} = y_1 y_2$. Then the variable y_{123} is computed by forming P_{123} using $\zeta = y_{12}$ within P_3 to obtain $y_{123} = y_1 y_2 y_3$. Continuing up to J_m, the variable y_{J_m} is computed by forming P_{J_m} using $\zeta = y_{J_{m-1}}$ within P_m to obtain $y_{J_m} = \prod_{j=1}^{m} y_j$. Here, each set P_{J_ℓ} has the same number $(2n_\ell + 1)$ of constraints and the same variables $\boldsymbol{u}_\ell$ as P_ℓ, but includes y_{J_ℓ} and $y_{J_{\ell-1}}$ instead of y_ℓ.

In the spirit of the above discussion, for each P_{J_ℓ} with $\ell \ge 2$, it is sufficient to have the adjusted $\underline{M}_{\ell j}$ and $\overline{M}_{\ell j}$, denoted $\underline{M}_{J_\ell j}$ and $\overline{M}_{J_\ell j}$ respectively, satisfy $\zeta f_\ell(\theta_{\ell j}) - \zeta f_\ell(x_\ell) \le \underline{M}_{J_\ell j}$ and $\zeta f_\ell(x_\ell) - \zeta f_\ell(\theta_{\ell j}) \le \overline{M}_{J_\ell j}$ for all possible realizations of $\zeta = y_{J_{\ell-1}} = \prod_{j=1}^{\ell-1} f_j(x_j)$ and $f_\ell(x_\ell)$. These values can be computed in various ways. One method is to have $\underline{M}_{J_\ell j} = \overline{M}_{J_\ell j} = f_{J_\ell}^+ - f_{J_\ell}^-$ where, as in Section 3, the terms $f_{J_\ell}^+$ and $f_{J_\ell}^-$ are upper and lower bounds on the product $\prod_{j=1}^{\ell} f_j(x_j)$. Different possibilities for these bounds exist. Again as in Section 3, if $f_j^- \ge 0$ for all $j \in \{1, \ldots, \ell\}$, then we can use $f_{J_\ell}^- = \prod_{j=1}^{\ell} f_j^-$ and $f_{J_\ell}^+ = \prod_{j=1}^{\ell} f_j^+$. If $f_j^- < 0$ for some j, then we can instead use $f_{J_\ell}^+ = \prod_{j=1}^{\ell} \max\{|f_j^-|, |f_j^+|\}$ and $f_{J_\ell}^- = -f_{J_\ell}^+$. Strengthened values for $\underline{M}_{J_\ell j}$ and $\overline{M}_{J_\ell j}$ can be computed based on problem structure and expended effort.

The size of the formulation is as follows. A count on each variable type is given in Table 2. Including the m original variables x_ℓ, there are $3m - 1$ continuous and $\sum_{\ell=1}^{m} \lceil \log_2(n_\ell) \rceil$ binary variables. Relative to constraints, each set P_ℓ for $\ell \in \{1, \ldots, m\}$ has $2n_\ell + 1$ restrictions and each set P_{J_ℓ} for $\ell \in \{2,\ldots,m\}$ has $2n_\ell$ additional restrictions. Also, $2n_\ell$ more inequalities are needed to handle the variables x_ℓ. The total number of constraints is then $m+4n_1+6\sum_{\ell=2}^{m} n_\ell$.

Table 2 Variable types and counts in Approach 1 of Li and Lu (2009).

Variable name	Variable type	Number of such variables
x_ℓ	continuous	m
y_ℓ	continuous	m
$y_{J_\ell}, \ell \neq 1$	continuous	$m-1$
$\boldsymbol{u}_\ell$	binary	$\lceil \log_2(n_\ell) \rceil$ for each $\ell \in \{1,\ldots,m\}$

4.2. Li & Lu Approach 2

The second approach of Li and Lu (2009) also represents functions of discrete variables, and their products, using logarithmic numbers of binary variables. For simplicity in presentation, we again begin by examining a single discrete variable $x \in S \equiv \{\theta_1,\theta_2,\ldots,\theta_n\}$ and function $f(x)$ so that we can temporarily drop the subscript ℓ.

While completely different in form and structure, this approach can be viewed as a blending of the first method of Li and Lu (2009) that makes use of the linear functions $A_j(\boldsymbol{u})$ of (20) for binary $\boldsymbol{u} \in \mathbb{R}^{\lceil \log_2 n \rceil}$ with our method that employs a vector of nonnegative, continuous variables $\boldsymbol{\lambda} \in \mathbb{R}^n$ summing to unity. It operates by creating a nonlinear equation in $\boldsymbol{\lambda}$ and $\boldsymbol{u}$ to enforce that $\boldsymbol{\lambda}$ is binary for $\boldsymbol{u}$ binary, and then sets $x=\theta_j$ and $y=f(\theta_j)$ for that single $\lambda_j=1$. The nonlinear equation is subsequently linearized using a result of Glover (1975). Notably, our study will show that the resulting formulation allows for a substantial simplification that is achieved by identifying inequalities that can be set to equality, removing extraneous variables, and deleting redundant constraints. These simplifications render both the functions $A_j(\boldsymbol{u})$ and the linearization of Glover (1975) wholly unnecessary. In fact, the restrictions of the simplified form are directly obtainable by multiplying the equations $V\boldsymbol{\lambda} = \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}$ of (6) found in $\bar{\Lambda}'$ by the invertible matrix B of (19), thus establishing an equivalence between the resulting sets.

To begin, recall from the first approach of Li and Lu (2009) in the previous section that the linear functions $A_j(\boldsymbol{u})$ of (20) were defined so that, for each $j \in \{1,\ldots,n\}$, $A_j(\boldsymbol{u})=0$ if $\boldsymbol{u}=\boldsymbol{v}_j$ and

$A_j(\boldsymbol{u}) \geq 1$ if $\boldsymbol{u} \neq \boldsymbol{v}_j$. Also recall for each such j that the vector $\boldsymbol{v}_j$ denotes the base-2 expansion of $j-1$, where entry i corresponds to the value 2^{i-1}. In this manner, $A_j(\boldsymbol{u})$ is defined for every binary $\boldsymbol{u}$ satisfying $\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n-1$. The second approach of Li and Lu (2009) defines a vector of nonnegative, continuous variables $\boldsymbol{\lambda} \in \mathbb{R}^n$ that is restricted to have $\sum_{j=1}^n \lambda_j = 1$, and uses the nonlinear equation $\sum_{j=1}^n A_j(\boldsymbol{u})\lambda_j = 0$ to ensure that the single $j \in \{1, \ldots, n\}$, say p, having $A_p(\boldsymbol{u}) = 0$ must also have $\lambda_p = 1$. Then the equations

$$x = \sum_{j=1}^{n} \theta_j \lambda_j \text{ and } y = \sum_{j=1}^{n} f(\theta_j)\lambda_j, \tag{24}$$

which are identical to those found in (5) and (9), enforce $x = \theta_p$ and $y = f(\theta_p)$. The system is below.

$$Q \equiv \left\{ \begin{array}{c} (\boldsymbol{u}, \boldsymbol{\lambda}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n : \\ \sum_{j=1}^n \lambda_j = 1, \\ \sum_{j=1}^n A_j(\boldsymbol{u})\lambda_j = 0, \\ \sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n-1, \\ \boldsymbol{u} \text{ binary}, \ \boldsymbol{\lambda} \geq \mathbf{0} \end{array} \right\}$$

The paper of Li and Lu (2009) linearizes the quadratic equation with the same method of Glover (1975) that was used to rewrite the nonlinear restrictions of (10) as (11). The first step is to factor the variables u_k from $\boldsymbol{\lambda}$. Expressing this factorization in terms of earlier notation, by (20) we obtain

$$\sum_{j=1}^{n} A_j(\boldsymbol{u})\lambda_j = \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}^T BV\boldsymbol{\lambda},$$

where the matrix B is as defined in (19). For each $k \in \{1, \ldots, \lceil \log_2 n \rceil + 1\}$, denoting the k^{th} row of the vector $BV\boldsymbol{\lambda}$ by $g_{k-1}(\boldsymbol{\lambda})$ so that

$$\begin{bmatrix} g_0(\boldsymbol{\lambda}) \\ \vdots \\ g_{\lceil \log_2 n \rceil}(\boldsymbol{\lambda}) \end{bmatrix} = BV\boldsymbol{\lambda}, \tag{25}$$

the equation $\sum_{j=1}^n A_j(\boldsymbol{u})\lambda_j = 0$ in Q becomes

$$g_0(\boldsymbol{\lambda}) + \sum_{k=1}^{\lceil \log_2 n \rceil} g_k(\boldsymbol{\lambda}) u_k = 0.$$

For each $k \in \{1, \ldots, \lceil \log_2 n \rceil\}$, the method of Glover (1975) substitutes a continuous variable δ_k for the product $g_k(\boldsymbol{\lambda})u_k$, and uses four inequalities to enforce $\delta_k = g_k(\boldsymbol{\lambda})u_k$ at binary $\boldsymbol{u}$. Using the fact that each such $g_k(\boldsymbol{\lambda})$ is lower and upper bounded by -1 and 1 respectively (since the coefficient on every λ_j in each function is $-1, 0$, or 1 and the sum of the λ_j equals 1), the formulation is as given below. The paper of Li and Lu (2009) does not include the restriction $\sum_{k=1}^{\lceil \log_2 n \rceil} 2^{k-1} u_k \leq n-1$ of Q; it can be shown redundant in the presence of the remaining constraints.

$$Q' \equiv \left\{ \begin{array}{ll} (\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n \times \mathbb{R}^{\lceil \log_2 n \rceil} : & \\ \quad \sum_{j=1}^{n} \lambda_j = 1 & \text{(26a)} \\ \quad g_0(\boldsymbol{\lambda}) + \sum_{k=1}^{\lceil \log_2 n \rceil} \delta_k = 0 & \text{(26b)} \\ \quad g_k(\boldsymbol{\lambda}) - (1-u_k) \leq \delta_k \leq g_k(\boldsymbol{\lambda}) + (1-u_k) \; \forall \, k = 1, \ldots, \lceil \log_2 n \rceil & \text{(26c)} \\ \quad -u_k \leq \delta_k \leq u_k \; \forall \, k = 1, \ldots, \lceil \log_2 n \rceil & \text{(26d)} \\ \quad \boldsymbol{u} \text{ binary}, \; \boldsymbol{\lambda} \geq \boldsymbol{0} & \end{array} \right\}$$

While not noted in Li and Lu (2009), the structure of Q' allows for a simplification that significantly reduces the numbers of variables and constraints. Consider the theorem below.

THEOREM 2. *Every point* $(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\delta}})$ *with* $\hat{\boldsymbol{\lambda}} \geq \boldsymbol{0}$ *and* $\boldsymbol{0} \leq \hat{\boldsymbol{u}} \leq \boldsymbol{1}$ *that satisfies (26a)–(26d) has* $-\hat{u}_k = \hat{\delta}_k = g_k(\hat{\boldsymbol{\lambda}}) - (1-\hat{u}_k)$ *for all* $k \in \{1, \ldots, \lceil log_2 n \rceil\}$.

Proof of Theorem 2. It is readily verified that the matrix B defined in (19) has the first row of B^{-1}, say $\boldsymbol{\rho}^T \in \mathbb{R}^{\lceil \log_2 n \rceil + 1}$, with $\frac{2}{\lceil \log_2 n \rceil}$ in the first entry and $\frac{1}{\lceil \log_2 n \rceil}$ elsewhere. Consequently,

$$\sum_{j=1}^{n} \lambda_j = \boldsymbol{\rho}^T B V \boldsymbol{\lambda} = \frac{2}{\lceil \log_2 n \rceil} g_0(\boldsymbol{\lambda}) + \frac{1}{\lceil \log_2 n \rceil} \sum_{k=1}^{\lceil \log_2 n \rceil} g_k(\boldsymbol{\lambda}), \tag{27}$$

where the first equality recognizes the first row of $V\boldsymbol{\lambda}$ from (6) as $\sum_{j=1}^{n} \lambda_j$, and the second equality follows from (25). Now, sum $\frac{2}{\lceil \log_2 n \rceil}$ times the equation in (26b) with $\frac{1}{\lceil \log_2 n \rceil}$ times the sum of the left inequalities in (26c) and (26d) and invoke (27) to obtain

$$\sum_{j=1}^{n} \lambda_j \leq 1. \tag{28}$$

But (26a) enforces this restriction with equality for all $(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\delta}) \in Q'$. Then the left inequalities of both (26c) and (26d) must also hold with equality for all $(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\delta}) \in Q'$. This completes the proof. □

The above theorem allows us to equivalently rewrite Q' with the left inequalities of (26c) and (26d) satisfied with equality so that $\delta_k = g_k(\boldsymbol{\lambda}) - (1-u_k)$ and $\delta_k = -u_k$ for each $k \in \{1, \ldots, \lceil \log_2 n \rceil\}$. This makes the right inequalities redundant due to $\boldsymbol{0} \leq \boldsymbol{u} \leq \boldsymbol{1}$. Then we can substitute $\delta_k = -u_k$

throughout the problem so that the variables $\boldsymbol{\delta}$ and restrictions (26d) are no longer needed. The resulting reduced version of Q' is RQ' below.

$$RQ' \equiv \left\{ \begin{array}{l} (\boldsymbol{u},\boldsymbol{\lambda}) \in \mathbb{R}^{\lceil \log_2 n \rceil} \times \mathbb{R}^n : \\ \qquad \sum_{j=1}^{n} \lambda_j = 1 \\ \qquad g_0(\boldsymbol{\lambda}) = \sum_{k=1}^{\lceil \log_2 n \rceil} u_k \\ \qquad g_k(\boldsymbol{\lambda}) = 1 - 2u_k,\ \forall\, k = 1,\ldots,\lceil \log_2 n \rceil \\ \qquad \boldsymbol{u} \text{ binary},\ \boldsymbol{\lambda} \geq \boldsymbol{0} \end{array} \right\}$$

Denoting the continuous relaxations of Q' and RQ' where the binary restrictions on $\boldsymbol{u}$ are replaced with $\boldsymbol{0} \leq \boldsymbol{u} \leq \boldsymbol{1}$ by $\bar{Q}'$ and $\bar{RQ}'$ respectively, it directly follows that a point $(\hat{\boldsymbol{u}},\hat{\boldsymbol{\lambda}},\hat{\boldsymbol{\delta}}) \in \bar{Q}'$ if and only if $\hat{\boldsymbol{\delta}} = -\hat{\boldsymbol{u}}$ and $(\hat{\boldsymbol{u}},\hat{\boldsymbol{\lambda}}) \in \bar{RQ}'$. Thus, $\bar{RQ}'$ can be viewed as an economical representation of $\bar{Q}'$ that is obtained by setting a subset of the inequalities to equality, and by removing redundant constraints and unnecessary variables.

The proof of Theorem 2 shows that RQ' can be further reduced in size by removing any one of the $\lceil \log_2 n \rceil + 2$ equality restrictions. This follows from (27), as each such restriction can be expressed as a linear combination of the others, with no multipliers of value 0.

Interestingly, the set $\bar{RQ}'$ provides exactly the same polyhedral region as $\bar{\Lambda}'$. This equivalence is addressed in the theorem below.

THEOREM 3. *A point* $(\hat{\boldsymbol{u}},\hat{\boldsymbol{\lambda}}) \in \bar{RQ}'$ *if and only if* $(\hat{\boldsymbol{u}},\hat{\boldsymbol{\lambda}}) \in \bar{\Lambda}'$.

Proof of Theorem 3. Multiply the restrictions $V\boldsymbol{\lambda} = \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}$ of $\bar{\Lambda}'$ by the invertible matrix B of (19). Then (25) and the structure of B gives that the equation $BV\boldsymbol{\lambda} = B\begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}$ yields the last $1 + \lceil \log_2 n \rceil$ equations found within $\bar{RQ}'$. As noted above, the restriction $\sum_{j=1}^{n} \lambda_j = 1$ is implied by the remaining equations of $\bar{RQ}'$, completing the proof. □

EXAMPLE 4. As in Example 3, let $f(x) = x$ with $x \in S = \{1,3,5\}$, so that again $n = 3$ with $\lceil \log_2 n \rceil = 2$. The set $\bar{\Lambda}'$ in three nonnegative continuous variables $\boldsymbol{\lambda}$, two binary variables $\boldsymbol{u}$, and three equality constraints is given in Example 3 where $V\boldsymbol{\lambda} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}$. By (25), $\begin{bmatrix} g_0(\boldsymbol{\lambda}) \\ g_1(\boldsymbol{\lambda}) \\ g_2(\boldsymbol{\lambda}) \end{bmatrix} = \begin{bmatrix} \lambda_2 + \lambda_3 \\ \lambda_1 - \lambda_2 + \lambda_3 \\ \lambda_1 + \lambda_2 - \lambda_3 \end{bmatrix} = BV\boldsymbol{\lambda}$ with $B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & -2 & 0 \\ 1 & 0 & -2 \end{bmatrix}$ so that the representation of Li and Lu (2009) using $\bar{Q}'$ is

$$\bar{Q}' = \left\{ \begin{array}{l} (\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\delta}) \in \mathbb{R}^2 \times \mathbb{R}^3 \times \mathbb{R}^2, \boldsymbol{\lambda} \geq \mathbf{0} : \\ \begin{array}{rcl} \lambda_1 + \lambda_2 + \lambda_3 & = 1 & \\ \lambda_2 + \lambda_3 + \delta_1 + \delta_2 & = 0 & \\ \lambda_1 - \lambda_2 + \lambda_3 - 1 + u_1 & \leq \delta_1 \leq & \lambda_1 - \lambda_2 + \lambda_3 + 1 - u_1 \\ \lambda_1 + \lambda_2 - \lambda_3 - 1 + u_2 & \leq \delta_2 \leq & \lambda_1 + \lambda_2 - \lambda_3 + 1 - u_2 \\ -u_1 & \leq \delta_1 \leq & u_1 \\ -u_2 & \leq \delta_2 \leq & u_2 \\ 0 & \leq u_1 \leq & 1 \\ 0 & \leq u_2 \leq & 1 \end{array} \end{array} \right\}.$$

Theorems 2 and 3 ensure that every point $(\boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\delta}) \in \bar{Q}'$ must have $\boldsymbol{\delta} = -\boldsymbol{u}$, and that a point $(\boldsymbol{u}, \boldsymbol{\lambda}) \in \bar{\Lambda}'$ if and only if $(\boldsymbol{u}, \boldsymbol{\lambda}, -\boldsymbol{u}) \in \bar{Q}'$. However, the form of $\bar{Q}'$ is larger than $\bar{\Lambda}'$. It uses the extra variables δ_1 and δ_2 and, not counting the lower bounds of 0 on u_1 and u_2, requires two equality and ten inequality constraints. To illustrate Theorem 2 that the four left inequalities restricting δ_1 and δ_2 must hold with equality, sum the second constraint with $\frac{1}{2}$ times each of these four inequalities to obtain $\lambda_1 + \lambda_2 + \lambda_3 \leq 1$, as (28) was computed from (27). The first equation of $\bar{Q}'$ then establishes the result. The representation of $f(x)$ (equivalently x for this example) is achieved using (24).

The paper of Li and Lu (2009) notes that this approach can be combined with their first method to handle products of univariate functions. Given m functions $f_\ell(x_\ell)$ where $x_\ell \in S_\ell \equiv \{\theta_{\ell 1}, \theta_{\ell 2}, \ldots, \theta_{\ell n_\ell}\}$ for $\ell \in \{1, \ldots, m\}$, the product $\prod_{\ell=1}^m f_\ell(x_\ell)$ is linearized in an identical fashion to the previous section with the following exception. For each $\ell \in \{1, \ldots, m\}$, a set Q'_ℓ in the variables $\boldsymbol{u}_\ell$, $\boldsymbol{\lambda}_\ell$, and $\boldsymbol{\delta}_\ell$ is formed as in (26a)–(26d) so that x_ℓ and $f_\ell(x_\ell)$ can be expressed as in (24). Then the representations Q'_ℓ replace the sets P_ℓ. For each $\ell \in \{2, \ldots, m\}$, the set P_{J_ℓ} remains unchanged, having the variable y_{J_ℓ} represent the product $\prod_{j=1}^{\ell} f_j(x_j)$.

Relative to the number of constraints, for each $\ell \in \{1, \ldots, m\}$ the set Q'_ℓ and the corresponding expressions in (24) contain $4\lceil \log_2(n_\ell) \rceil + 4$ restrictions (noting that $\mathbf{0} \leq \boldsymbol{u} \leq \mathbf{1}$ is implied). For $\ell \in \{2, \ldots, m\}$ the set P_{J_ℓ} has $2n_\ell$ additional restrictions. In total, $4m + 4\sum_{\ell=1}^m \lceil \log_2(n_\ell) \rceil + 2\sum_{\ell=2}^m n_\ell$ constraints are required. (This is a savings beyond the first method in Li and Lu (2009) of $4n_\ell - 4\lceil \log_2(n_\ell) \rceil - 3$ constraints for each $\ell \in \{1, \ldots, m\}$.) As for variables, Table 3 gives the names, types, and numbers required. Summing, there are $3m - 1 + \sum_{\ell=1}^m (n_\ell + \lceil \log_2(n_\ell) \rceil)$ continuous and $\sum_{\ell=1}^m \lceil \log_2(n_\ell) \rceil$ binary variables.

As a final remark here, it is important to note that two works subsequent to Li and Lu (2009) focus on the methods of this subsection and the previous, and that our contribution of Section 2

Table 3 Variable types and counts in Approach 2 of Li and Lu (2009).

Variable name	Variable type	Number of such variables
x_ℓ	continuous	m
y_ℓ	continuous	m
$y_{J_\ell}, \ell \neq 1$	continuous	$m-1$
$\boldsymbol{\lambda}_\ell$	continuous	n_ℓ for each $\ell \in \{1,\ldots,m\}$
$\boldsymbol{\delta}_\ell$	continuous	$\lceil \log_2(n_\ell) \rceil$ for each $\ell \in \{1,\ldots,m\}$
$\boldsymbol{u}_\ell$	binary	$\lceil \log_2(n_\ell) \rceil$ for each $\ell \in \{1,\ldots,m\}$

answers an open question in this regard. The paper of Li et al. (2009) applied these works of Li and Lu (2009) to piecewise-linear functions, and showed that the second is preferable to the first. However, Vielma et al. (2010a) later demonstrated, both theoretically and computationally, that the forms given within Li et al. (2009) are dominated by alternate methods. Vielma et al. (2010a) conjectured that, while inferior in this setting, the second might have utility in other contexts. Theorems 2 and 3 shed light in this regard by establishing an equivalence between the polytopes $\bar{Q}'$ and $\bar{\Lambda}'$ so that $\bar{Q}'$ is locally ideal. This equivalence reveals two shortcomings of the second form of Li et al. (2009). First, the set $\bar{Q}'$ can be reduced in size to $\bar{\Lambda}'$ without forfeiting relaxation strength. Second, and more importantly, the weak relaxation relative to piecewise functions is not due to the set $\bar{Q}'$ itself, but rather to the specific modeling of the functions employed by Li et al. (2009).

4.3. Computational Comparisons

Subsections 4.1 and 4.2 demonstrate that the proposed approach improves upon the methods of Li and Lu (2009), but it remains to show the extent of this improvement within a computational setting. To provide insight, this subsection presents numerical experience for a simple nonlinear discrete program in three variables as described below.

$$\text{minimize} \quad c_1 x_1^{p_1} + c_2 x_2^{p_2} + c_3 x_3^{p_3} + c_{12} x_1^{p_1} x_2^{p_2} + c_{13} x_1^{p_1} x_3^{p_3} + c_{23} x_2^{p_2} x_3^{p_3} + c_{123} x_1^{p_1} x_2^{p_2} x_3^{p_3}$$

subject to

$$0.9 \leq x_1^{p_1} x_2^{p_2} x_3^{p_3} \leq 1.1$$

$$x_\ell \in S_\ell \equiv \{\theta_{\ell 1}, \dots, \theta_{\ell n}\} \quad \forall\, \ell \in \{1, \dots, 3\}$$

Here, the decision variables are given by x_1, x_2, and x_3, with n denoting the number of possible realizations of each x_ℓ (so that by earlier notation, $n_\ell = n$ for $\ell \in \{1,2,3\}$). Each of the seven nonempty subsets J of $\{1,2,3\}$ has the objective coefficient c_J corresponding to the term $\prod_{\ell \in J} x_\ell^{p_\ell}$ generated randomly via a uniform distribution on the open interval $(-1,1)$. For each $\ell \in \{1,2,3\}$, the exponents p_ℓ are similarly obtained, while the values $\theta_{\ell j}$ also follow a uniform distribution, but over the interval $(0,100)$. All input was truncated to two decimal places. In a manner similar to test problems of Li and Lu (2009), the constraints bounding the product $x_1^{p_1} x_2^{p_2} x_3^{p_3}$ were chosen to restrict the feasible region beyond that of the sets S_ℓ to make the problems more challenging. For simplicity of presentation, we assume without loss of generality that the elements of each set S_ℓ are arranged in increasing order so that $\theta_{\ell 1} < \dots < \theta_{\ell n}$. (No scenarios are considered where, for some ℓ, two $\theta_{\ell j}$ are the same since the problem could be trivially reduced in size.)

To build our formulation, begin by constructing the set Λ'_ℓ for each $\ell \in \{1,2,3\}$ to model the function $f_\ell(x_\ell) = x_\ell^{p_\ell}$ as in (16). Then for $\ell \in \{1,2\}$, lower and upper bounds f_ℓ^- and f_ℓ^+ on the function $f_\ell(x_\ell)$ are given by $f_\ell^- = \theta_{\ell n}^{p_\ell}$ and $f_\ell^+ = \theta_{\ell 1}^{p_\ell}$ if $p_\ell < 0$, and by $f_\ell^- = \theta_{\ell 1}^{p_\ell}$ and $f_\ell^+ = \theta_{\ell n}^{p_\ell}$ if $p_\ell > 0$. These values provide bounds on the product $f_1(x_1) f_2(x_2)$ as $f_{12}^- = f_1^- f_2^-$ and $f_{12}^+ = f_1^+ f_2^+$ respectively. Consistent with (16)–(18), define nonnegative κ_2 and κ_3 by $\kappa_2 = f_1(x_1) - f_1^-$ and $\kappa_3 = f_1(x_1) f_2(x_2) - f_{12}^-$ so that $\kappa_2^- = \kappa_3^- = 0$, $\kappa_2^+ = f_1^+ - f_1^-$, and $\kappa_3^+ = f_{12}^+ - f_{12}^-$. Then use (17) and (18) with $\ell \in \{2,3\}$ to have $\Gamma'_2(\kappa_2)$ and $\Gamma'_3(\kappa_3)$ set $y_{12} = f_1(x_1) f_2(x_2)$ and $y_{123} = f_1(x_1) f_2(x_2) f_3(x_3)$. To model $f_1(x_1) f_3(x_3)$ and $f_2(x_2) f_3(x_3)$, repeat the logic of (17) and (18) to create new sets, say $\Gamma'_3(\kappa_2)$ and $\Gamma'_3(\kappa_4)$, by scaling the restrictions defining Λ'_3 by each of κ_2 and $\kappa_4 = f_2(x_2) - f_2^-$ so that we can use $y_{13} = f_1(x_1) f_3(x_3)$ and $y_{23} = f_2(x_2) f_3(x_3)$. Here, $\kappa_4^- = 0$ and $\kappa_4^+ = f_2^+ - f_2^-$. The net effect is to have $y_1 = x_1^{p_1}$, $y_2 = x_2^{p_2}$, $y_3 = x_3^{p_3}$, $y_{12} = x_1^{p_1} x_2^{p_2}$, $y_{13} = x_1^{p_1} x_3^{p_3}$, $y_{23} = x_2^{p_2} x_3^{p_3}$, and $y_{123} = x_1^{p_1} x_2^{p_2} x_3^{p_3}$.

We compare the two approaches of Li and Lu (2009) with our method of Section 3. All formulations were submitted to CPLEX 11 with default optimization parameters on a Sun V440 workstation having 16GB of RAM and four 1.6GHz processors. A time limit of 30 minutes per problem was enforced. Results are summarized in Tables 4 and 5. Table 4 is separated into four main columns, delineated by vertical lines. Column 1 gives the numbers of realizations n of each variable x_ℓ, while columns 2, 3, and 4 consider approaches 1 and 2 of Li and Lu (2009), and our approach respectively. Within each of these last three columns, the CPU execution times in seconds, the numbers of dual-simplex iterations, and the numbers of nodes explored in the binary search tree to reach optimality are labeled "Time," "Iters," and "Nodes," respectively. (Problems recognized as infeasible were not recorded.) As indicated in column 1, the number of realizations of each variable x_ℓ was increased from 32 to 64 to 128 to 256, running five problems of each size.

Table 4 Comparison of computational performance

	Li & Lu Approach 1			Li & Lu Approach 2			Our Approach		
n	Time	Iters	Nodes	Time	Iters	Nodes	Time	Iters	Nodes
32	0.41	1057	427	0.17	357	50	0.01	180	0
32	1.34	2693	854	0.24	536	106	0.08	218	0
32	2.37	6813	2426	0.46	1249	254	0.17	947	46
32	11.98	51238	23819	1.56	7619	1623	0.21	1220	107
32	13.81	63192	26057	2.78	28637	2430	0.15	596	40
64	6.63	12181	4409	3.82	14275	2292	0.43	1372	185
64	10.11	19788	6540	4.79	14487	3434	0.39	1629	97
64	11.93	25457	5868	2.23	7645	443	0.44	1878	162
64	17.02	31970	16291	0.58	1010	113	0.40	779	23
64	26.27	59311	18290	1.10	1360	224	0.18	230	15
128	50.00	49237	16172	10.46	11712	3503	1.22	1443	73
128	>1800.00	-	-	8.09	8312	1743	0.36	654	39
128	>1800.00	-	-	13.24	7838	2843	0.97	2703	100
128	>1800.00	-	-	56.99	108090	25161	1.06	4458	207
128	>1800.00	-	-	58.51	130196	24007	1.18	3641	430
256	297.99	120309	41283	236.69	371169	40509	0.87	485	37
256	>1800.00	-	-	39.21	55242	4305	1.46	3113	80
256	>1800.00	-	-	42.51	39389	3898	1.38	3005	55
256	>1800.00	-	-	484.75	358470	95583	0.61	448	11
256	>1800.00	-	-	631.01	612257	66601	2.87	7811	299

The results confirm the superiority of the proposed method. In all cases, our method outperformed both approaches of Li and Lu (2009) in terms of time, iterations, and numbers of nodes explored. This advantage becomes more pronounced as the problem sizes increase. Also, a comparison of columns 2 and 3 reaffirms the experience of Li and Lu (2009) that their second approach is an improvement over their first.

Table 5 addresses relaxation strength for the same twenty test problems. The five columns denote the numbers of realizations n of each variable x_ℓ, the objective values to the relaxations of approaches 1 and 2 of Li and Lu (2009), to our relaxations, and to that of the original binary program. In every case, our method yields a tighter bound than the second approach of Li and Lu (2009), which in term is tighter than the first approach. Interestingly, while Theorems 2 and 3 prove relaxation equivalence between our method and the second approach *for a single function*, this equivalence does not apply to products of more than one function since, as explained in Subsection 4.2, the authors revert back to their first approach to model functional products. Thus, the second approach of Li and Lu (2009) has a theoretical relaxation strength between their first approach and ours.

Table 5 Comparison of relaxation strength

n	Li & Lu Approach 1	Li & Lu Approach 2	Our Approach	Binary Optimal
32	-8.342	-4.033	-0.697	-0.697
32	-12.645	0.241	3.398	3.538
32	-18.675	-10.211	-0.681	1.121
32	-363.184	-355.244	-51.507	2.583
32	-1119.101	-971.884	-12.023	-5.727
64	-0.132	0.828	2.543	3.785
64	-121.358	-41.941	-36.386	-34.129
64	-169.248	-71.218	-7.053	-2.886
64	-12.744	-3.701	-0.724	0.253
64	-58.522	-37.552	-4.186	-2.241
128	-3.304	-1.217	-0.218	2.128
128	-112.355	-83.511	-0.162	0.082
128	-123.285	-100.147	-14.434	-10.826
128	-115.093	-85.041	-5.721	0.719
128	-6159.661	-4935.720	-401.697	-32.183
256	-2.604	-0.206	0.331	1.235
256	-15.314	-10.119	-2.925	-1.866
256	-42.301	-31.235	-8.338	-5.400
256	-315.553	-72.972	-9.926	-7.098
256	-339.325	-273.020	-30.575	-8.910

5. Conclusions

This paper presents a strategy for expressing functions of discrete variables, and their products, in terms of logarithmic numbers of binary variables. The fundamental idea is an observation for writing a binary vector as a convex combination of extreme points of the unit hypercube. This observation allows us to treat n binary variables as continuous by defining a smaller number of $\lceil \log_2 n \rceil$ binary variables. Such collections of binary variables naturally arise in modeling general discrete variables, and functions thereof.

Our strategy provides a unifying perspective for two published approaches that are designed to use logarithmic numbers of binary variables. It compares favorably, in terms of the strengths of the continuous relaxations and formulation sizes, to both methods. We show for the case of a function $f(x)$ having x a discrete variable, that our continuous relaxation dominates one such method, and is theoretically equivalent to the other. For both competing approaches, our forms use markedly fewer constraints. Our proofs provide insight into relationships of the alternate approaches with each other, and improve upon the second by identifying (previously unnoticed) families of unnecessary constraints and extraneous variables. The established theoretical superiority is then borne out for products of functions in a computational study.

Given a collection of m functions $f_\ell(x_\ell)$ for $\ell \in \{1, \dots, m\}$, where each discrete variable x_ℓ can realize n_ℓ distinct values, Table 6 summarizes the numbers of continuous variables and constraints required to linearize the product $\prod_{\ell=1}^{m} f_\ell(x_\ell)$ for each of the three approaches. The first row of the table is the proposed method of Section 3, while rows two and three are the approaches of Sections 4.1 and 4.2. For readability, we let $N = \sum_{\ell=1}^{m} n_\ell$ and $L = \sum_{\ell=1}^{m} \lceil \log_2(n_\ell) \rceil$. Since *all three* approaches employ the same L binary variables, this count is not included in the table.

We also posed four reduction strategies based on variable substitutions and transformations. In order to perform more transparent comparisons, these strategies are not reflected in Table 6. However, it is interesting to note that, in addition to the proposed method, they can be selectively applied to the other two approaches. The substitution of variables $\boldsymbol{w}'_\ell = \boldsymbol{w}_\ell - \kappa_\ell^- \boldsymbol{u}_\ell$ in the first strategy for positive κ_ℓ^- is applicable to the second approach of Li and Lu (2009), although it becomes unnecessary in light of Theorem 2. The second reduction strategy to eliminate the variables x_ℓ and y_ℓ is applicable to the second approach of Li and Lu (2009). But all variables in the first approach of Li and Lu (2009), and the y_{J_ℓ} in the second approach, must be kept. The third reduction strategy that converts equality restrictions to inequalities can be applied to the second approach of Li and Lu (2009), but will only save two variables, due to only two equality restrictions. Finally, the fourth reduction strategy dealing with the order of the functions considered can potentially reduce all formulations, though to different extents.

Table 6 Summary of variable and constraint counts

	Continuous Variables	Constraints
Proposed Method	$3m-1-n_1+2N+L-\lceil\log_2(n_1)\rceil$	$5m-2-5\lceil\log_2(n_1)\rceil+6L$
Li & Lu 1 Li and Lu (2009)	$3m-1$	$m+6N-2n_1$
Li & Lu 2 Li and Lu (2009)	$3m-1+N+L$	$4m+4L+2N-2n_1$

This study is predominantly theoretical in nature, focusing on representation size and relaxation strength, as well as equivalences between, and improvements to, known techniques. Limited computational experience is provided. Future research includes detailed studies to more fully determine the practical benefits made possible by reduced numbers of binary variables in concise model representations.

Acknowledgments

The authors are grateful to the National Science Foundation for partially supporting this research under grant number CMMI-0968909. We also thank two anonymous referees for their insightful comments and suggestions.

References

Adams, W.P., R.J. Forrester. 2005. A simple recipe for concise mixed 0-1 linearizations. *Operations Research Letters* **33**(1) 55–61.

Bergamini, M.L., N.J. Scenna, P.A. Aguirre. 2007. Global optimal structures of heat exchanger networks by piecewise relaxation. *Industrial & Engineering Chemistry Research* **46**(6) 1752–1763.

Boyd, S.P., S.J. Kim, D.D. Patil, M.A. Horowitz. 2005. Digital circuit optimization via geometric programming. *Operations Research* **53**(6) 899–932.

Floudas, C.A., C.E. Gounaris. 2009. A review of recent advances in global optimization. *Journal of Global Optimization* **45**(1) 3–38.

Glover, F. 1975. Improved linear integer programming formulations of nonlinear integer programs. *Management Science* **22**(4) 455–460.

Glover, F. 1984. An improved MIP formulation for products of discrete and continuous variables. *Journal of Information and Optimization Sciences* **5**(1) 469–471.

Harjunkoski, I., T. Westerlund, R. Pörn. 1999. Numerical and environmental considerations on a complex industrial mixed integer non-linear programming (MINLP) problem. *Computers & Chemical Engineering* **23**(10) 1545–1561.

Kallrath, J. 2003. Exact computation of global minima of a noncovex portfolio optimization problem. *Frontiers in Global Optimization*, Kluwer Academic Publishers, 237–254.

Li, H.L., H.C. Lu. 2009. Global optimization for generalized geometric programs with mixed free-sign variables. *Operations Research* **57**(3) 701–713.

Li, H.L., H.C. Lu, C.H. Huang, N.Z. Hu. 2009. A superior representation method for piecewise linear functions. *INFORMS Journal on Computing* **21**(2) 314–321.

Meyer, C.A., C.A. Floudas. 2006. Global optimization of a combinatorially complex generalized pooling problem. *American Institute of Chemical Engineers Journal* **52**(3) 1027–1037.

Sahinidis, N.V., M. Tawarmalani. 2000. Applications of global optimization to process and molecular design. *Computers & Chemical Engineering* **24**(9-10) 2157–2169.

Sinha, M., L.E.K. Achenie, G.M. Ostrovsky. 1999. Environmentally benign solvent design by global optimization. *Computers & Chemical Engineering* **23**(10) 1381–1394.

Vielma, J.P., S. Ahmed, G.L. Nemhauser. 2010a. A Note on "A Superior Representation Method for Piecewise Linear Functions." *INFORMS Journal on Computing* **22**(3) 493–497.

Vielma, J.P., S. Ahmed, G.L. Nemhauser. 2010b. Mixed-integer models for nonseparable piecewise-linear optimization: unifying framework and extensions. *Operations Research* **58**(2) 303–315.

Vielma, J.P., G.L. Nemhauser. 2011. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming* **128**(1-2) 49–72.

Watters, L.J. 1967. Reduction of integer polynomial programming problems to zero-one linear programming problems. *Operations Research* **15**(6) 1171–1174.