ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

Modeling without Categorical Variables: A Mixed-Integer Nonlinear Program for the Optimization of Thermal Insulation Systems [1]

Kumar Abhishek, Sven Leyffer, and Jeffrey T. Linderoth

Mathematics and Computer Science Division

Preprint ANL/MCS-P1434-0607

June 21, 2007

[1] This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. This work was also supported by the U.S. Department of Energy through the grant DE-FG02-05ER25694.

Contents

Modeling without Categorical Variables: A Mixed-Integer Nonlinear Program for the Optimization of Thermal Insulation Systems[*]

KUMAR ABHISHEK[†], SVEN LEYFFER[‡], AND JEFFREY T. LINDEROTH[§]

June 21, 2007

Abstract

Optimal design applications are often modeled by using categorical variables to express discrete design decisions, such as material types. A disadvantage of using categorical variables is the lack of continuous relaxations, which precludes the use of modern integer programming techniques. We show how to express categorical variables with standard integer modeling techniques, and we illustrate this approach on a load-bearing thermal insulation system. The system consists of a number of insulators of different materials and intercepts that minimize the heat flow from a hot surface to a cold surface. Our new model allows us to employ black-box modeling languages and solvers and illustrates the interplay between integer and nonlinear modeling techniques. We present numerical experience that illustrates the advantage of the standard integer model.

Keywords: Mixed integer nonlinear programming, modeling with binary variables, thermal insulation systems, categorical variables.

AMS-MSC2000: 90C11, 90C30, 90C90

1 Introduction

Recently, researchers have expressed interest in mixed-variable optimization problems (MVPs). Problems of this class involve *categorical variables*, which are constrained to take values from a finite set of non-numerical values. MVPs have been used to design load-bearing thermal insulation systems, where the categorical variables model the type of material chosen for the insulators (Kokkolaras et al., 2001; Abramson,

[*] Preprint ANL/MCS-P1434-0607.

[†] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, kua3@lehigh.edu.

[‡] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, leyffer@mcs.anl.gov.

[§] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, jtl3@lehigh.edu.

2004). Categorical variables are a convenient way to move from a simulation tool that requires input such as material properties to an optimization tool. On the other hand, the presence of categorical variables precludes the use of modern integer optimization techniques, because continuous relaxations of the categorical decisions are not readily available.

We show how categorical variables can be replaced by standard integer modeling techniques. We illustrate our approach on an example of thermal insulation systems and emphasize the interaction of integer and nonlinear modeling techniques. Our approach provides a blueprint for reformulating other design problems that involve categorical variables, for example the design of nanomaterials (Zhao et al., 2005), and in optimal sensor placement (Beal et al., 2006). We believe that the conclusions of this paper also are relevant to application scientists who develop simulation tools. In our view it is important to include optimization considerations in simulation tools from the start.

This paper is organized as follows. We start by reviewing the categorical variable formulation of a thermal insulation system. Next, we introduce the integer and nonlinear modeling techniques that allow us to reformulate this model as a standard mixed integer nonlinear programming (MINLP) model. We obtain three models with varying degree of smoothness and comment on the relative merits of these formulations. Numerical experiments illustrating the benefit of our new approach are presented.

2 Load-Bearing Thermal Insulation Design

We consider the design of a load-bearing thermal insulation system. This system uses a series of heat intercepts and insulators to minimize the heat flow from a hot surface to a cold surface. The objective is to minimize the power required to maintain the heat intercepts at certain temperatures so that the cold surface can be maintained at the required temperature; see Figure 1. The insulator types are chosen from a set $\mathcal{M}$ of materials and are modeled as categorical variables.

The model is described in detail by Abramson (2004), who extends the model given by Kokkolaras et al. (2001) by adding load-bearing requirements. Thus, the insulators act as mechanical supports and must satisfy certain load-bearing constraints involving quantities such as thermal expansion, system mass, and stress.

2.1 Model Parameters and Data

The parameters and data of the model are summarized in Table 1.

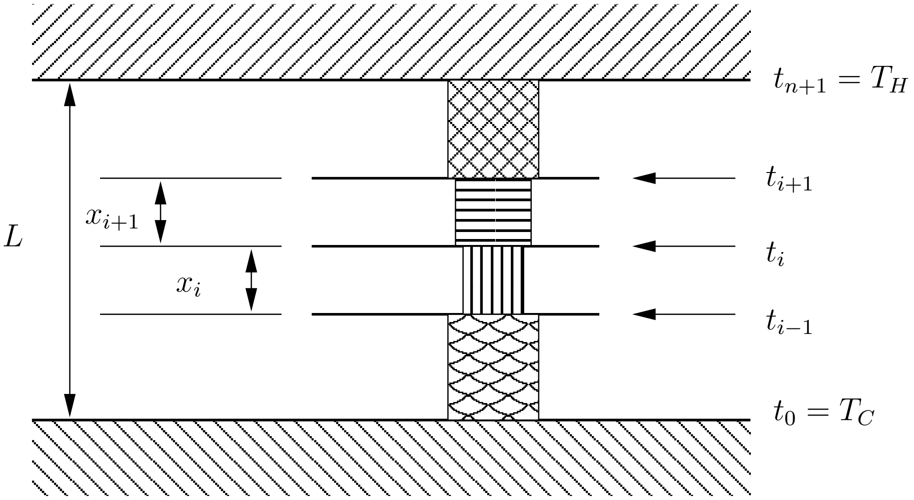

Figure 1: Illustration of the thermal insulation system.

Table 1: Model Parameters and Data

Parameter	Description	Value in Case Study Instance
$C(t_i)$	thermodynamic cycle efficiency of intercept i, see (2.1)	
$e(t,m)$	thermal expansion of insulator $m \in \mathcal{M}$ at temperature t	
F	system load	250 kN
$k(t,m)$	thermal conductivity of insulator $m \in \mathcal{M}$ at temperature t	
L	system length	10 cm
M	maximum system mass	10 kg
$\mathcal{M}$	set of insulator materials	see Table 2
N	maximum number of intercepts	10
T_C	cold surface temperature	4.2K
T_H	hot surface temperature	300K
δ	maximum thermal expansion	5%
$\rho(m)$	density of insulator $m \in \mathcal{M}$	see Table 2
$\sigma(t,m)$	tensile yield strength of insulator $m \in \mathcal{M}$ at temperature t	

The thermodynamic cycle efficiency of intercept i is a piecewise constant function of the temperature:

$$C(t_i) = \begin{cases} 5 & \text{if } t_i \leq 4.2K, \\ 4 & \text{if } 4.2K < t_i < 71K \\ 2.5 & \text{if } t_i \geq 71K. \end{cases}, \quad i = 1, \ldots, n \tag{2.1}$$

The types of insulators are nylon, Teflon, epoxy(normal), epoxy(plane), 6063-T5 aluminum, 1020 low-carbon steel, and 304 stainless steel. Their corresponding densities, $\rho(m)$, are given in Table 2. Data for the thermal conductivity, $k(t, m)$, tensile yield strength, $\sigma(t, m)$, and thermal expansion, $e(t, m)$ are given in the form of look-up tables for every material $m \in \mathcal{M}$ and a discrete set of temperature values τ_j. Abramson (2004) and Kokkolaras et al. (2001) have fitted cubic splines to the data to provide a smooth approximation of these functions for every type of material. Our tables are also made available electronically as AMPL (Fourer et al., 2003) data files.

Table 2: Densities for the Various Insulator Materials

Nylon	Teflon	Epoxy-normal	Epoxy-plane	Aluminum	Steel	Carbon-steel
0.0010	0.0015	0.0018	0.0018	0.0027	0.0078	0.0078

2.2 Model Variables

We summarize the definition of the model variables in Table 3. Throughout, we index the intercepts by subscripts $i = 0, \ldots, n + 1$, where index $i = 0$ corresponds to the cold surface and index $i = n + 1$ corresponds to the hot surface. The material types are indexed by subscripts $j = 1, \ldots, |\mathcal{M}|$.

Table 3: Model Variables

Variable	Description
a_i	area of insulator $i = 1, \ldots, n + 1$
m_i	material $m_i \in \mathcal{M}$ of insulator $i = 1, \ldots, n + 1$
n	number of intercepts, $n \in \{1, 2, \ldots, N\}$
q_i	heat flow from intercept i to $i - 1$, for $i = 1, \ldots, n + 1$
t_i	temperature at intercept $i = 0, \ldots, n + 1$
Δx_i	thermal expansion of layer $i = 1, \ldots, n + 1$

Power is applied at each intercept i at its cooling temperature $t_i, i = 1, \ldots, n$. The gap between the intercepts $i - 1$ and i is filled in with insulator of thickness x_i. The temperature of the hot surface is given by $t_{n+1} = T_H$ and that of the cold surface by $t_0 = T_C$. The insulators used in the system may have different cross-sectional areas a_i. The design of the system involves choosing the number of intercepts n, their cooling temperatures t_i, the insulator types m_i, their thickness x_i and the cross-sectional areas a_i. We include the thermal expansion, $\Delta x_i / x_i$ for convenience but note that it is later eliminated from the model. The presence of categorical variables such as insulator types m_i and the number of insulators n make the model into a mixed-variable program.

2.3 Mixed-Variable Optimization Model

We can now state the complete mixed-variable optimization model.

$$\text{minimize} \quad \sum_{i=1}^{n} C(t_i) \left(\frac{T_H}{t_i} - 1 \right) \cdot (q_{i+1} - q_i) \tag{2.2}$$

$$\text{subject to} \quad q_i = \frac{a_i}{x_i} \int_{t_{i-1}}^{t_i} k(t, m_i) dt, \quad i = 1, \ldots, n+1 \tag{2.3}$$

$$\sum_{i=1}^{n} \rho(m_i) a_i x_i \leq M \tag{2.4}$$

$$\frac{F}{a_i} \leq \bar{\sigma}_i = \min\{\sigma(t, m_i) \,:\, t_{i-1} \leq t \leq t_i\}, \quad i = 1, \ldots, n+1 \tag{2.5}$$

$$\sum_{i=1}^{n} \left(\frac{\Delta x_i}{x_i} \right) \left(\frac{x_i}{L} \right) \leq \frac{\delta}{100} \tag{2.6}$$

$$\frac{\Delta x_i}{x_i} = \frac{\int_{t_{i-1}}^{t_i} e(t, m_i) k(t, m_i) dt}{\int_{t_{i-1}}^{t_i} k(t, m_i) dt}, \quad i = 1, \ldots, n \tag{2.7}$$

$$\sum_{i=1}^{n} x_i = L \tag{2.8}$$

$$t_{i-1} \leq t_i \leq t_{i+1}, \quad i = 1, \ldots, n \tag{2.9}$$

$$t_0 = T_C \quad \text{and} \quad t_{n+1} = T_H \tag{2.10}$$

$$x_i \geq 0, \quad a_i \geq 0, \quad i = 1, \ldots, n+1 \tag{2.11}$$

The model contains five classes of nonlinear constraints. Equation (2.3) defines the heat flow from intercept i to $i-1$, which is governed by Fourier's law. Equation (2.4) is the mass constraint of the system. The stress of the system must not exceed the specified load F, which is modeled by Equation (2.5). The thermal expansion constraint is modeled by Equation (2.6), with the thermal contraction $\frac{\Delta x_i}{x_i}$ defined by the constraint (2.7). In addition, the model contains some linear constraints: (2.8) constrains the thickness of the design, (2.9) orders the temperatures, and (2.10) fixes the temperatures at the cold and hot surface. We note that the latter two constraints imply that $T_C \leq t_i \leq T_H$ for $i = 0, \ldots, N+1$.

Abramson (2004) uses a different objective function in his `matlab` implementation, namely,

$$f_2 := \sum_{i=1}^{n+1} q_i \left(C(t_{i-1})(\frac{T_H}{t_{i-1}} - 1) - C(t_i)(\frac{T_H}{t_i} - 1) \right). \tag{2.12}$$

We use this objective function from now on for the purpose of modeling and for comparing our results with the work of Abramson.

The integrals in (2.3) and (2.7) are approximated by Simpson's rule, where the material specific functions $e(t, m_i)$ and $k(t, m_i)$ are derived from cubic spline interpolation of tabulated data.

Solving the thermal insulation problem involves evaluating a nonlinear objective function and nonlinear constraints over a variable space that includes categorical variables. The model is solved by using a pattern-search algorithm (Audet and Dennis, 2004, 2000; Abramson et al., 2004). The computational burden of pattern-search techniques grows with the number of variables, and this fact motivates the removal of as many defined variables as possible. For example, the fixed variables t_0 and t_{n+1} are removed. We can also remove the thermal expansion variables $\frac{\Delta x_i}{x_i}$ by substituting (2.7) into (2.6). In addition, it is argued in (Abramson, 2004) that the stress constraint must be binding at a solution, thereby implying that $a_i = \frac{F}{\bar{\sigma}_i}$ and allowing us to remove the variables a_i.

2.4 Challenges of the MVP Model

The MVP model (2.2)–(2.10) introduces a number of difficulties that appear to make it impossible to employ standard MINLP techniques to solve the problem:

1. The model contains a number of categorical variables that do not allow continuous relaxations. For example, it is not clear how to relax the condition that the materials be chosen from $\mathcal{M}$. Worse, the variable n appears as the upper limit in the summation, which makes constraints such as (2.8), (2.4), (2.6), and the objective function ((2.2) or (2.12)) discontinuous. Moreover, by changing n, we also change the number of variables t_i and so forth that appear in the model. In this sense, every n defines a different model.

2. There exists no analytic closed-form expression for the integrals in (2.3) and (2.7). Instead, the integrals are evaluated by using Simpson's rule. Thus, derivatives are difficult to compute, and hence derivative-free optimization techniques are used.

3. Constraint (2.5) contains a minimization for which no closed-form expression exists. The presence of this constraint results in a bilevel optimization problem that is considerably harder to solve. We note

that this constraint can be written equivalently as an infinite set constraint, by requiring that

$$\frac{F}{a_i} \le \sigma_i(t, m_i), \quad \forall t \in [t_{i-1}, t_i],\ \forall i = 1, \ldots, n+1.$$

4. The objective function is discontinuous because of the presence of the thermodynamic cycle efficiency coefficients, $C(t_i)$; see (2.1). This discontinuity can cause derivative-based NLP solvers to fail.

Each of these points is a potential death-blow for standard optimization techniques. Worse, even though pattern-search techniques can be applied, the discontinuities imply that it is almost impossible to verify the optimality of a solution returned by the pattern-search method.

Another drawback of pattern-search techniques is the fact that the quality of the solution depends on the definition of the neighborhood for the search. It is not hard to construct examples where the MVP pattern-search (Audet and Dennis, 2000) fails to find the minimum. Consider

$$\text{minimize } q(x) := (s_1(cx_1 - sx_2))^2 + (s_2(sx_1 - cx_2))^2 \quad \text{subject to } \ x_1, x_2 \ \text{ integer},$$

where, for example, $s_1 = 8, s_2 = 1$ are scaling parameters and $c = \cos(\pi/8),\ s = \sin(\pi/8)$ are rotational parameters. Figure 2 illustrates this situation: the blue points all correspond to pattern-optimal solutions, yet clearly only one of them corresponds to the local/global minimum of $q(x)$.

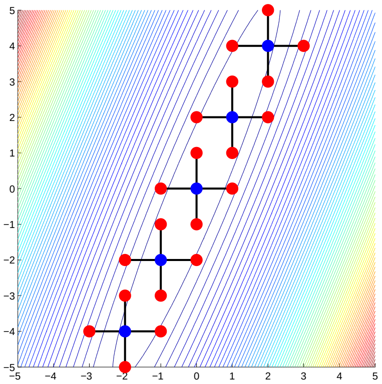

Figure 2: Example illustrating failure of MVP pattern-search.

Next, we show how each of these challenges can be tackled by combining integer and nonlinear modeling techniques. These reformulations result in a standard MINLP that we formulate in the modeling language AMPL (Fourer et al., 2003). The new formulation allows us to derive cutting planes and employ more powerful optimization techniques. In addition, we believe that by using a modeling language, our model becomes more transparent, ultimately enabling the design of larger and more complex systems.

3 Modeling Categorical Variables with Binary Variables

We start by showing that the categorical variables can be replaced by integer variables, and we develop a derivative-free model of thermal insulation that allows relaxations to be computed. We use binary indicator variables y_i to remove the variable n (the number of intercepts) from the summation bound in (2.4) and (2.6) and to eliminate the dependence of the number of decision variables on the decision variable n; we use binary decision variables y_i to denote the existence of layer i. This is a natural reformulation and not necessarily inefficient in practice, since the maximum number of intercepts N is typically small. For the specific instance we solve in Section 6, $N = 20$. This reformulation is made by introducing the following inequality system:

$$\begin{aligned}\sum_{i=1}^{N+1} y_i &= n+1 && (3.13)\\ y_{i+1} &\le y_i, \qquad i=1,\ldots,N && (3.14)\\ x_i &\le Ly_i \qquad i=1,\ldots,N+1 && (3.15)\\ y_i &\in \{0,1\}, \qquad i=1,\ldots,N+1. &&\end{aligned}$$

The inequalities (3.14) order the intercepts and ensure that only consecutive intercepts in this ordering can exist. The variable upper bound inequalities (3.15) ensure that there is a positive thickness to the layer only if the layer is chosen to exist. The inequalities (3.15) can be replaced by the much stronger set of inequalities

$$\sum_{j=i}^{N+1} x_j \le Ly_i \qquad i=1,\ldots,N+1. \tag{3.16}$$

In fact, Theorem 3.1 establishes that the inequalities (3.16) are facets of the convex hull of the inequality system of the reformulation.

Theorem 3.1. *Let*

$$P = \text{conv}\left(\{(x,y) \in \mathbb{R}_+^{N+1} \times \mathbb{B}_+^{N+1} \mid y_{i+1} \le y_i \quad i=1,\ldots,N, \quad x_i \le Ly_i \quad i=1,\ldots,N+1\}\right).$$

Then

$$\sum_{j=i}^{N+1} x_j \le Ly_i$$

defines a facet of P for each $i = 1,2,\ldots,N+1$.

Proof. Assume that for each $i = 1,\ldots,N+1$ there is an inequality $\pi^i x + \mu^i y \le \pi_0^i$ that is valid for P and satisfies

$$F_i \overset{\text{def}}{=} \{(x,y) \in P \mid \sum_{j=i}^{N+1} x_j = Ly_i\} \subseteq \{(x,y) \in P \mid \pi^i x + \mu^i y = \pi_0^i\} \overset{\text{def}}{=} \hat{F}_i.$$

In this case, we will show that $\pi^i x + \mu^i y \leq \pi_0^i$ is a scalar multiple of the inequality $\sum_{j=i}^{N+1} x_j \leq Ly_i$ which implies that $\sum_{j=i}^{N+1} x_j \leq Ly_i$ is a facet-defining inequality for P. (See (Nemhauser and Wolsey, 1988), Theorem I.4.3.5 for a proof of this result.)

First note that for each $i = 1, \ldots, N+1$, the point $(0,0) \in F_i$, so also the origin $(0,0) \in \hat{F}_i$, which implies that $\pi^i(0) + \mu^i(0) = \pi_0^i$, or

$$\pi_0^i = 0 \quad \forall i = 1, \ldots, N+1. \tag{3.17}$$

For any $i \geq 2$, the point $(0, e_1) \in F_i$. Since this point is also in $\hat{F}_i$, we have that $\mu_1^i = \pi_0^i = 0 \; \forall i \geq 2$. For any $i \geq 3$, the point $(0, e_1 + e_2) \in F_i$, so $\mu_1^i + \mu_2^i = \pi_0^i \; \forall i \geq 3$. Similarly, one can establish that

$$\mu_k^i = 0 \quad \forall i = 2, \ldots, N+1, \; \forall k \leq i-1. \tag{3.18}$$

For every $i = 1, \ldots, N+1$ the point $(Le_i, \sum_{j=1}^{i} e_j) \in F_i$. This implies that for each i, $L\pi_i^i + \sum_{j=1}^{i} \mu_j^i = \pi_0^i$, but by (3.18) and (3.17), this implies that

$$\mu_i^i = -L\pi_i^i \quad \forall i = 1, \ldots, N+1. \tag{3.19}$$

For any $i = 2, \ldots, N+1$ and for any $k = 1, \ldots, i-1$, the point

$$\left(Le_i + \sum_{j=1}^{k} Le_j, \sum_{j=1}^{i} e_j \right) \in F_i,$$

which implies that

$$L \sum_{j=1}^{k} \pi_j^i + L\pi_i^i + \sum_{j=1}^{i} \mu_j^i = 0.$$

This, coupled with (3.18) and (3.17), implies that

$$\sum_{j=1}^{k} \pi_j^i = 0 \quad \forall i = 2, \ldots, N+1, \; k = 1, \ldots, i-1. \tag{3.20}$$

Using (3.20) sequentially for $k = 1, 2, \ldots, i-1$, one can see that

$$\pi_k^i = 0 \quad \forall i = 2, \ldots, N+1, \; k = 1, \ldots, i-1. \tag{3.21}$$

For any $i = 1, \ldots, N$ and for any $k = i+1, \ldots, N+1$, the following three points lie on the face F_i:

- $(Le_i, \sum_{j=1}^{i} e_j)$,
- $(Le_k, \sum_{j=1}^{k} e_j)$, and
- $(L/2e_i, L/2e_k, \sum_{j=1}^{k} e_j)$.

Since these points also then must lie on $\hat{F}_i$ and the relations (3.18) and (3.17) hold, this fact implies that the equations

$$\begin{aligned} L\pi_i^i + \mu_i^i &= 0 \\ L\pi_k^i + \sum_{j=1}^{k} \mu_j^i &= 0 \\ L/2\pi_i^i + L/2\pi_k^i + \sum_{j=1}^{k} \mu_j^i &= 0 \end{aligned}$$

hold, which in turn implies that

$$\pi_k^i = \pi_i^i \quad \forall i = 1, \ldots, N,\ k = i+1, \ldots, N+1 \tag{3.22}$$

$$\sum_{j=i+1}^{k} \mu_j^i = 0 \quad \forall i = 1, \ldots, N,\ k = i+1, \ldots, N+1. \tag{3.23}$$

Using the relation (3.23) with $k = i + 1$, one can establish that $\mu_{i+1}^i = 0$. Then, using (3.23) subsequently with $k = i + 2, \ldots N + 1$, one can establish that

$$\mu_k^i = 0 \quad \forall i = 1, \ldots, N,\ k = i+1, \ldots, N+1. \tag{3.24}$$

Collecting the relations (3.17), (3.18), (3.19), (3.21), (3.22), and (3.18), we see that indeed the inequality $\pi^i x + \mu^i y \leq \pi_0^i$ is a scalar multiple of the inequality $\sum_{j=i}^{N+1} x_j \leq L y_i$ for every $i = 1, 2, \ldots, N + 1$, which completes the proof. □

Having indicator variables y_i representing the existence of layer i also makes it convenient to model material properties. We let $z_{ij} = 1$ if and only if the j^{th} material is chosen for layer i (where the ordering of the material is arbitrary). Otherwise, we set $z_{ij} = 0$. We can use the following constraints to model the fact that only existing layers choose a material type.

$$\sum_{j=1}^{|\mathcal{M}|} z_{ij} = y_i, \quad i = 1, \ldots, N+1. \tag{3.25}$$

The constraints (2.8), (2.4), and (2.6) that involve n as a summation limit can now be written equivalently by using the new summation bound N and the indicator variables y_i.

Next, we model the constraints involving data functions such as thermal conductivity $k(t, m)$ by observing that

$$k(t, m_i) = \sum_{j=1}^{|\mathcal{M}|} z_{ij} k(t, m_j),$$

where (with some abuse of notation) m_i is the material used in layer i, while m_j is the j^{th} material in $\mathcal{M}$. We can now remove the categorical variables from the mixed-variable model and define the following mixed-integer model:

$$\text{minimize} \quad \sum_{i=1}^{N+1} q_i y_i \left(C(t_{i-1})(\frac{T_H}{t_{i-1}} - 1) - C(t_i)(\frac{T_H}{t_i} - 1) \right) \; := \; CoolingPower \tag{3.26}$$

$$\text{subject to} \quad q_i = \frac{a_i}{x_i} \int_{t_{i-1}}^{t_i} \sum_{1}^{|\mathcal{M}|} z_{ij} k(t, m_j) dt, \quad i = 1, \dots N+1 \tag{3.27}$$

$$\sum_{i=1}^{N+1} \sum_{j=1}^{|\mathcal{M}|} \rho_j z_{ij} a_i x_i \le M, \tag{3.28}$$

$$\frac{F}{a_i} \le \bar{\sigma}_i = \min_t \left\{ \sum_{j=1}^{|\mathcal{M}|} z_{ij} \sigma(t, m_j) \; : \; t_{i-1} \le t \le t_i \right\}, \quad i = 1, \dots, n+1 \tag{3.29}$$

$$\sum_{i=1}^{N+1} \left(\frac{\Delta x_i}{x_i} \right) x_i \le L \frac{\delta}{100}, \tag{3.30}$$

$$\frac{\Delta x_i}{x_i} = \frac{\int_{t_{i-1}}^{t_i} \sum_{j=1}^{|\mathcal{M}|} z_{ij} e(t, m_j) k(t, m_j) dt}{\int_{t_{i-1}}^{t_i} \sum_{j=1}^{|\mathcal{M}|} z_{ij} k(t, m_j) dt}, \quad i = 1, \dots, N \tag{3.31}$$

$$\sum_{i=1}^{N+1} x_i = L, \tag{3.32}$$

$$x_i \ge \epsilon L y_i, \quad i = 1, \dots, N+1, \tag{3.33}$$

$$\sum_{j=i}^{N+1} x_i \le L y_i, \quad i = 1, \dots, N+1, \tag{3.34}$$

$$t_i - t_{i-1} \ge \epsilon y_i, \quad i = 1 \dots N+1 \tag{3.35}$$

$$t_0 = T_C \tag{3.36}$$

$$t_i \ge T_H \cdot (1 - y_{i+1}), \quad i = 1, \dots, N \tag{3.37}$$

$$A_{\text{min}} y_i \le a_i \le A_{\text{max}} y_i, \quad i = 1 \dots N+1, \tag{3.38}$$

where $\epsilon > 0$ is a small constant and ρ_j is the density of the j^{th} material.

In addition to the constraints (3.34), which ensure that the width of layer i, $x_i = 0$, is zero whenever that layer does not exist, we add a variable lower bound (3.33). Equation (3.37) fixes the temperatures t_i of the layers that do not exist. This constraint ensures that for the nonexisting layers $j \ge n+1$, the term in the objective function is canceled, because $T_H/t_j - 1 = 0$, for all $j \ge n+1$ ($t_j = T_H$). In addition, we model the condition that the temperatures at the intercepts are nondecreasing ($t_i \ge t_{i-1}$) by insisting that they are separated by at least a small positive amount (again this helps avoid spurious solutions) by Equation (3.35).

Finally, the absence of a layer is modeled with (3.38), which sets the area a_i to zero, for any nonexisting layer, $i \geq n+2$.

Our model is a mixed-integer simulation model, owing to the presence of the inner minimization and the integrals. Below we show that we can further refine this model by developing a fully explicit MINLP model.

4 A MINLP Model for Thermal Insulation Design

In this section we show that the simulation model of the previous section can be formulated as a smooth MINLP that can be expressed by using standard modeling tools such as AMPL.

4.1 Avoiding Bilevel Optimization Problems

We start by showing that the structure of the data allows us to avoid the bilevel constraints (2.5). It is argued in (Abramson, 2004) that the relationship between a_i and the power applied at an intercept i (given as $C(t_i)\left(\frac{T_H}{t_i}-1\right)\cdot(q_{i+1}-q_i)$) imply that the constraints (2.5) are always binding and, thus, the area variables are removed from the model. We are not sure that this argument holds in general; instead, we proceed in a different way.

We introduce a finer piecewise linear approximation of the data $\sigma(t,m)$ by adding data points from the cubic spline interpolation used by Abramson (2004), denoted by

$$T_C = T_1 < T_2 < \ldots < T_D = T_H, \tag{4.1}$$

where D is the number of discretization points (typically 20). Next, we approximate the minimization in (2.5) by requiring that the bound hold at temperatures at each intercept. Thus we introduce the constraints

$$Fz_{ij} \leq a_i\left(\sigma(T_r,m_j) + \frac{\sigma(T_{r+1},m_j)-\sigma(T_r,m_j)}{T_{r+1}-T_r}(t_i - T_r)\right), \tag{4.2}$$

for all materials $j = 1,\ldots,|\mathcal{M}|,\ i = 1,\ldots,N+1$, where the index r is such that $T_r \leq t_i < T_{r+1}$. We note that for most materials, $\sigma(t,m)$ is monotonic in t, and we need to enforce this bound only at the upper end of each intercept. However, the epoxy materials do not have monotonic $\sigma(t,m)$, and we must therefore enforce the bounds at both the lower and the upper intercept. We do this by adding the constraints at t_{i-1}

$$Fz_{ij} \leq a_i\left(\sigma(T_r,m_j) + \frac{\sigma(T_{r+1},m_j)-\sigma(T_r,m_j)}{T_{r+1}-T_r}(t_{i-1} - T_r)\right), \tag{4.3}$$

for all materials $m_j \in \{\text{epoxy-n}, \text{epoxy-p}\}$ and $i = 1,\ldots,N+1$, where the index k is such that $T_r \leq t_{i-1} < T_{r+1}$. The presence of the conditional statement involving t_{i-1} complicates these constraints.

A convenient way to model the conditional relationship $T_r \leq t_{i-1} \leq T_{r+1}$ in (4.2) is as the following summation.

$$Fz_{ij} \leq a_i \sum_{\substack{r=1 \\ T_r \leq t_{i-1} \leq T_{r+1}}}^{D} \left(\sigma(T_r, m_j) + \frac{\sigma(T_{r+1}, m_j) - \sigma(T_r, m_j)}{T_{r+1} - T_r}(t_i - T_r) \right). \tag{4.4}$$

Similarly, we replace (4.3) by

$$Fz_{ij} \leq a_i \leq \sum_{\substack{r=1 \\ T_r \leq t_{i-1} \leq T_{r+1}}}^{D} \left(\sigma(T_r, m_j) + \frac{\sigma(T_{r+1}, m_j) - \sigma(T_r, m_j)}{T_{r+1} - T_r}(t_{i-1} - T_r) \right). \tag{4.5}$$

We note that only a single term in this summation will be active. The resulting constraint is continuous but not smooth, as t_i passes through the breakpoints. However, this nonsmoothness does not appear to cause any problems for the NLP solvers. In Section 5 we provide a reformulation of these nonsmooth constraints that employs integer variables and a finer discretization of T_r.

4.2 Evaluation of Integrals

The model involves integrals over the data functions $k(t, m)$ and $k(t, m) \cdot e(t, m)$ in (2.3) and (2.7). In (Abramson, 2004) these integrals are evaluated by using Simpson's rule, which is consistent with the piecewise cubic spline interpolation of the data. However, Simpson's rule adds nonlinearity and would be difficult to implement in a modeling language. Instead, we add more data points consistent with the cubic spline interpolation as in (4.1), and we evaluate the integrals using the trapezoidal rule on the data points

$$(T_r, k(T_r, m_j)) \quad \text{and} \quad (T_r, k(T_r, m_j) \cdot e(T_r, m_j)) \,.$$

We note that adding more data points does not increase the number of variables in the model and greatly reduces the nonlinearity of the constraints, without sacrificing accuracy, as can be seen from Figure 3, which shows the cubic spline versus the piecewise linear approximation of $k(t, \text{epoxy-p})$. The additional data points clearly improve the fidelity of the piecewise linear approximation, as can be seen from the dotted line that represents a piecewise linear interpolation of the original data points.

In general, the temperatures at the intercepts, t_i will not take the values used in the discretization, T_r, which we must take into account when calculating the values of the integrals. Figure 4 illustrates our approach. For a given integration range $[t_{i-1}, t_i]$, we split the integral into three distinct areas (A, B, and D in Figure 4) depending on the relative position of the variables t_{i-1}, t_i and the discretization points T_r. This

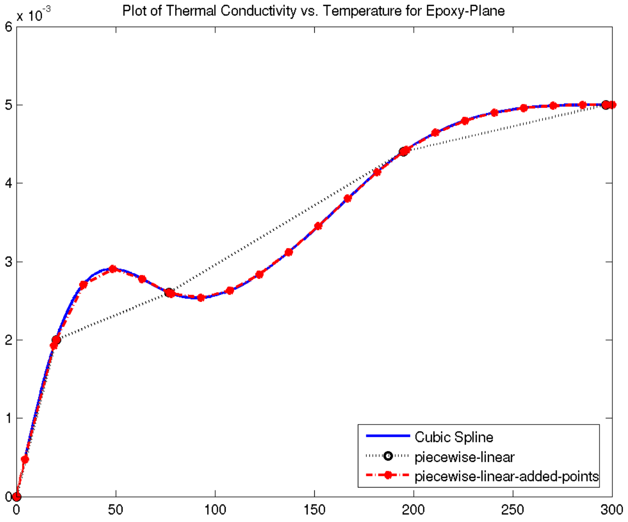

Figure 3: Cubic spline versus piecewise linear approximation of $k(t, \text{epoxy-p})$.

partition leads to the following disjoint index sets:

$$\begin{aligned}
\mathcal{B} &:= \{1 \le r \le D-1 : [T_r, T_{r+1}] \subset [t_{i-1}, t_i]\} && (B)\\
\mathcal{A} &:= \{1 \le r \le D-1 : t_{i-1} \in (T_r, T_{r+1}) \text{ and } t_i \ge T_{r+1}\} && (A)\\
\mathcal{D} &:= \{1 \le r \le D-1 : t_i \in (T_r, T_{r+1}) \text{ and } t_{i-1} \le T_r\} && (D)\\
\mathcal{E} &:= \{1 \le r \le D-1 : (T_r, T_{r+1}) \supset [t_{i-1}, t_i]\} && (E),
\end{aligned} \tag{4.6}$$

where the set $\mathcal{E}$ corresponds to the case where $[t_{i-1}, t_i]$ falls entirely into a single discretization interval. In each case, we approximate the integrals using the trapezoidal rule. We note that in case (B), we can precompute this approximation as parameters

$$K_{r,j} \approx \int_{t=T_r}^{T_{r+1}} k(t, m_j)dt \text{ and } E_{r,j} \approx \int_{t=T_r}^{T_{r+1}} k(t, m_j)e(t, m_j)dt$$

for each interval $[T_r, T_{r+1}]$ and for all materials m_j.

Next, we introduce notation to denote the trapezoidal approximations in cases (A), (D), and (E). For some function $g(t, m_j)$ we denote

$$A(g, m_j, r, t_{i-1}) := \frac{1}{2}\left(g(T_{r-1}, m_j) + \frac{g(T_r, m_j) - g(T_{r-1}, m_j)}{T_r - T_{r-1}}(t_{i-1} - T_{r-1}) + g(T_r, m_j)\right)(T_r - t_{i-1}),$$

$$D(g, m_j, r, t_i) := \frac{1}{2}\left(g(T_{r+1}, m_j) + \frac{g(T_r, m_j) - g(T_{r+1}, m_j)}{T_r - T_{r+1}}(T_{r+1} - t_i) + g(T_r, m_j)\right)(T_{r+1} - t_i),$$

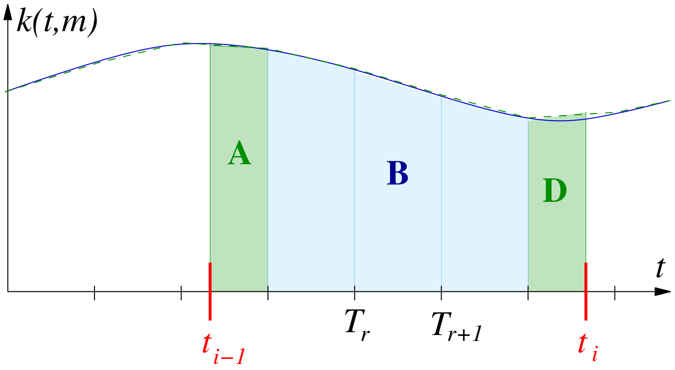

Figure 4: Illustration of the integral computation.

and

$$\begin{aligned}E(g, m_j, r, t_i) := \tfrac{1}{2} \quad &\Big(g(T_r, m_j) + \tfrac{g(T_{r+1}, m_j) - g(T_r, m_j)}{T_{r+1} - T_r}(t_{i-1} - T_r) \\ &+ g(T_r, m_j) + \tfrac{g(T_{r+1}, m_j) - g(T_r, m_j)}{T_{r+1} - T_r}(t_i - T_r) + g(T_r, m_j) \Big) (T_i - t_{i-1})\end{aligned}$$

as the trapezoidal approximation in the areas identified by the sets $\mathcal{A}$, $\mathcal{D}$, and $\mathcal{E}$, respectively. Introducing variables v_{ij} and w_{ij} that approximate

$$v_{ij} \approx \int_{t_{i-1}}^{t_i} k(t, m_j) dt, \quad w_{ij} \approx \int_{t_{i-1}}^{t_i} k(t, m_j) e(t, m_j) dt,$$

we can express v_{ij} and w_{ij} as

$$v_{ij} = \sum_{r \in \mathcal{B}} K_{r,j} + \sum_{r \in \mathcal{A}} A(k, m_j, r, t_{i-1}) + \sum_{r \in \mathcal{D}} D(k, m_j, r, t_i) + \sum_{r \in \mathcal{E}} E(k, m_j, r, t_i) \tag{4.7}$$

and

$$w_{ij} = \sum_{r \in \mathcal{B}} E_{r,j} + \sum_{r \in \mathcal{A}} A(k \cdot e, m_j, r, t_{i-1}) + \sum_{r \in \mathcal{D}} D(k \cdot e, m_j, r, t_i) + \sum_{r \in \mathcal{E}} E(k \cdot e, m_j, r, t_i). \tag{4.8}$$

The introduction of v_{ij}, w_{ij} allows us to formulate the load-bearing thermal insulation design MVP as a standard MINLP. Before proceeding, however, we simplify some of the nonlinear expressions further to avoid division by zero that can confound standard NLP software. Thus, we rewrite (2.3) as

$$q_i x_i = a_i \sum_{j=1}^{|\mathcal{M}|} z_{ij} v_{ij}, \quad i = 1, \ldots, N+1. \tag{4.9}$$

We introduce a new variable $u_i = \Delta x_i / x_i$ to model the thermal expansion of each layer, and we rewrite (2.7) as

$$u_i \sum_{j=1}^{|\mathcal{M}|} z_{ij} v_{ij} = \sum_{j=1}^{|\mathcal{M}|} z_{ij} w_{ij}, \quad i = 1, \ldots, N+1. \tag{4.10}$$

Thus, we can write (3.30) as

$$\sum_{i=1}^{N+1} u_i x_i \leq L\frac{\delta}{100}. \tag{4.11}$$

This reformulation removes the categorical variables $m \in \mathcal{M}$ from the model and leaves us with a standard MINLP given as follows.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{N+1} q_i y_i \left(C(t_{i-1})(\frac{T_H}{t_{i-1}} - 1) - C(t_i)(\frac{T_H}{t_i} - 1) \right) \;:=\; CoolingPower \\ \text{subject to} & (3.13), (3.14), (3.16), (3.25), (3.32), (3.28), (3.33) \\ & (3.37), (3.35), (3.38), (4.9), (4.10), (4.11) \\ & (4.4), (4.7), (4.8), \quad \forall j = 1, \ldots, |\mathcal{M}|,\; i = 1, \ldots, N+1 \\ & (4.5), \quad \forall m_j \in \{\text{epoxy-n}, \text{epoxy-p}\},\; i = 1, \ldots, N+1 \\ & t_0 = T_C \quad \text{and} \quad t_{N+1} = T_H \\ & y_i \in \{0,1\}, \quad \forall i = 1, \ldots, N+1 \\ & z_{ij} \in \{0,1\}, \quad \forall i = 1, \ldots, N+1,\; j = 1, \ldots, |\mathcal{M}| \\ & x_i, q_i, u_i, v_{ij}, w_{ij}, n, t_i, a_i \in \mathbb{R} \end{array} \tag{P-0}$$

We note that this model contains discontinuous objective coefficients. Next, we show how to reformulate these discontinuous objective coefficients to arrive at a smoother MINLP.

4.3 Modeling a Discontinuous Function with Binary Variables

The discontinuous thermodynamic cycle efficiency coefficient $C(t_i)$ in (2.1) can be replaced by a smooth relationship. We start by introducing additional binary variables $s_{ki} \in \{0,1\},\; k = 1, \ldots, 3$ to model the following implications.

$$t_i \leq 4.2 \Rightarrow\; s_{1i} = 1 \tag{4.12}$$

$$4.2 < t_i < 71 \Rightarrow\; s_{2i} = 1 \tag{4.13}$$

$$t_i \geq 71 \Rightarrow\; s_{3i} = 1 \tag{4.14}$$

Letting $\epsilon > 0$ be a small constant that models the strict inequalities in (2.1), we can model the implications (4.12) and (4.14) as

$$t_i - (T_C - 4.2 - \epsilon)s_{1i} \;\geq\; 4.2 + \epsilon \tag{4.15}$$

$$t_i - (T_H - 71 + \epsilon)s_{3i} \;\leq\; 71 - \epsilon, \tag{4.16}$$

respectively. Condition (4.13) is equivalent to

$$s_{2i} = 0 \;\Rightarrow\; t_i \le 4.2 \;\text{ or }\; t_i \ge 71,$$

which we model as follows:

$$\begin{aligned} s_{1i} + s_{2i} + s_{3i} &= 1 && (4.17)\\ t_i + (T_H - 4.2)s_{1i} &\le T_H && (4.18)\\ t_i - (71 - T_C)s_{i3} &\ge T_C. && (4.19) \end{aligned}$$

Equation (4.17) models the implication that each t_i is in exactly one interval. The two inequalities (4.18) and (4.19) fix $s_{ik} \in \{0,1\}$ given any t_i, and constrain t_i for any valid choice of $s_{ik} \in \{0,1\}$. The approach presented here is fairly general and applies to other piecewise functions as well.

4.4 A Piecewise Smooth MINLP Model

The smooth MINLP model is now given as follows.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{N+1} q_i y_i \left((5s_{1,i-1} + 4s_{2,i-1} + 2.5s_{3,i-1}) \left(\frac{T_H}{t_{i-1}} - 1 \right) \right. & \\ & \left. -(5s_{1i} + 4s_{2i} + 2.5s_{3i}) \left(\frac{T_H}{t_i} - 1 \right) \right) \;:=\; CoolingPower & \\ \text{subject to} & (3.13), (3.14), (3.16), (3.25), (3.32), (3.28), (3.33) & \\ & (3.37), (3.35), (3.38), (4.9), (4.10), (4.11) & \\ & (4.7), (4.8), (4.4), \quad \forall j = 1, \ldots, |\mathcal{M}|,\; i = 1, \ldots, N+1 & \\ & (4.5), \quad \forall m_j \in \{\text{epoxy-n}, \text{epoxy-p}\},\; i = 1, \ldots, N+1 & \text{(P-1)} \\ & (4.15), (4.16), (4.17), (4.18), (4.19), \quad \forall i = 1, \ldots, N+1 & \\ & t_0 = T_C \quad \text{and} \quad t_{N+1} = T_H & \\ & y_i \in \{0,1\}, \quad \forall i = 1, \ldots, N+1 & \\ & z_{ij} \in \{0,1\}, \quad \forall i = 1, \ldots, N+1,\; j = 1, \ldots, |\mathcal{M}|, & \\ & s_{ki} \in \{0,1\}, \quad \forall i = 1, \ldots, N+1,\; k = 1, \ldots, 3 & \\ & x_i, q_i, u_i, v_{ij}, w_{ij}, n, t_i, a_i \in \mathbb{R} & \end{array}$$

When we ran this model, we noticed that the objective function $CoolingPower$ can become negative, which is a nonphysical solution. More specifically, solving a relaxation of (P-1) at a node in a branch-and-bound procedure can yield a negative solution. The reason for this behavior is that the variables s_{ki} now have fractional values, and thus the differences

$$\left((5s_{1,i-1}+4s_{2,i-1}+2.5s_{3,i-1})\left(\frac{T_H}{t_{i-1}}-1\right)-(5s_{1i}+4s_{2i}+2.5s_{3i})\left(\frac{T_H}{t_i}-1\right)\right) \quad \forall i=1,\ldots,N+1 \tag{4.20}$$

need not be non-negative. Thus we add the constraint $CoolingPower >= 0$ to the models.

We note that model (P-1) is nonsmooth as a result of the presence of conditional statements in constraints (4.4), (4.5), (4.7), and (4.8), which include summations that depend on the temperature t_i. These nonsmooth constraints may cause trouble for standard NLP solvers. We show in Section 6 that we can solve such models despite the presence of nonsmooth equations.

5 A Smooth MINLP Model with Discretized Temperature

In this section we describe an alternative formulation of the thermal insulation model that assumes that we select the temperatures at the intercepts, t_i from a discrete set (4.1). This formulation may at first sight seem more complicated, but it allows us to remove the nonsmoothness present in model (P-1). We envisage using a large number of discretization points (e.g. at one-degree level). This assumption may seem strong. However, we note that we can always use the discrete values of t_i to discover the optimal value of the structure variables, n and $m_i \in \mathcal{M}$ and then run an NLP to adjust the temperatures and thickness of the final design. This simplification is justified by the fact that the solution of a MINLP is typically more sensitive to the choice of the integer variables than to the choice of the continuous variables.

As before, we model the discrete choice of temperatures by introducing SOS-1 variables,

$$t_i=\sum_{r=1}^{D} d_{ir}T_r, \quad 1=\sum_{r=1}^{D} d_{ir}, \quad d_{ir}\in\{0,1\}, \quad \forall i=0,\ldots,N+1, \tag{5.1}$$

where $d_{ir}=1$ if intercept i is kept at temperature T_r, and 0 otherwise. We note that the new variables d_{ir} are a SOS-1, and so we need to go only to at most $\log(D)$ branching levels in the tree, making this an efficient reformulation.

Introducing a discrete set of temperatures simplifies the MINLP model in a number of ways. We start by simplifying constraints (4.7) and (4.8). We precompute the integrals of $k(t,m_j)$ and $k(t,m_j)e(t,m_j)$ over $[T_1,T_r]$ as fixed parameters:

$$V_{rj} \;\approx\; \int_{T_1}^{T_r} k(t,m_j)dt \quad \text{and} \quad W_{rj} \;\approx\; \int_{T_1}^{T_r} k(t,m_j)e(t,m_j)dt$$

for all $r=1,\ldots,D$ and $j=1,\ldots,|\mathcal{M}|$. Next, we use the identity

$$v_{ij} \;=\; \int_{t=t_{i-1}}^{t_i} k(t,m_j)dt \;=\; \int_{t=T_1}^{t_i} k(t,m_j)dt - \int_{t=T_1}^{t_{i-1}} k(t,m_j)dt$$

and observe that (5.1) implies

$$\int_{t=T_1}^{t_i} k(t,m_j)dt = \sum_{r=1}^{D} d_{ir} \int_{t=T_1}^{T_r} k(t,m_j)dt.$$

Thus, equations (4.7) and (4.8) simplify to the following set of *linear* equations:

$$v_{ij} = \sum_{r=1}^{D} d_{ir}V_{rj} - \sum_{r=1}^{D} d_{i-1,r}V_{rj} \quad \forall i = 0,\ldots,N,\ j = 1,\ldots,|\mathcal{M}|, \tag{5.2}$$

$$w_{ij} = \sum_{r=1}^{D} d_{ir}W_{rj} - \sum_{r=1}^{D} d_{i-1,r}W_{rj} \quad \forall i = 0,\ldots,N,\ j = 1,\ldots,|\mathcal{M}|. \tag{5.3}$$

Next, we show how (5.1) simplifies the objective function. First, we observe that

$$\left(\frac{T_H}{t_i} - 1\right) \;=\; \left(\frac{T_H}{\sum_{r=1}^{D} d_{ir}T_r} - 1\right) \;=\; \left(\sum_{r=1}^{D} d_{ir}\frac{T_H}{T_r} - 1\right). \tag{5.4}$$

We can replace the binary variables s_{ik} modeling the discontinuous objective coefficients by defining constants

$$\hat{C}_r := \begin{cases} 5 & \text{if } T_r \le 4.2 \\ 4 & \text{if } 4.2K < T_r < 71K \\ 2.5 & \text{if } T_r \ge 71K \end{cases}, \tag{5.5}$$

which replace $C(t_i)$ in the objective function. Combining (5.4) and (5.5), we reformulate the objective function as

$$\sum_{i=1}^{N+1} q_i \left(\sum_{r=1}^{D}(d_{i-1,r} - d_{ir})\hat{C}_r\left(\frac{T_H}{T_r} - 1\right)\right) \;:=\; CoolingPower.$$

The constraints that bound the temperature difference between consecutive layers, (3.35), can be expressed as binary inequalities between d_{ir} and $d_{i-1,r}$:

$$\sum_{r=1}^{D} rd_{ir} \ge \sum_{r=1}^{D} rd_{i-1,r} + y_i, \quad \forall i = 1,\ldots,N+1, \tag{5.6}$$

which implies that $t_i > t_{i-1}$ by at least one discretization difference. Similarly, we can express constraints (3.37) as a discrete set of constraints:

$$d_{iD} \ge (1 - y_{i+1}), \quad \forall i = 1,\ldots,N. \tag{5.7}$$

We next simplify the area stress constraints (4.4) and (4.5). We start by rewriting the bilevel constraint (2.5) as an infinite set of constraints:

$$F \le a_i\sigma(t,m_i), \quad \forall t \in [t_{i-1},t_i], \quad \forall i = 1,\ldots,N+1.$$

We observe that the discrete range of the temperature variable allows us to formulate this constraint as a finite set of inequalities:

$$F \leq a_i \sigma(T_s, m_i), \quad \forall s : t_{i-1} \leq T_s \leq t_i, \quad \forall i = 1, \ldots, N+1.$$

We introduce the tensile yield strength $\sigma_{sj} := \sigma(T_s, m_j)$ of each material at all discrete temperature values to remove the dependence on the categorical variable m_i:

$$F \leq a_i \sum_{j=1}^{|\mathcal{M}|} z_{ij} \sigma_{sj}, \quad \forall s : t_{i-1} \leq T_s \leq t_i, \quad \forall i = 1, \ldots, N+1.$$

We formulate the condition $s : t_{i-1} \leq T_s \leq t_i$ using the SOS variables d_{ir} as follows. First, we introduce the following partial sums for notational convenience:

$$p_{is} := \sum_{r=1}^{s} d_{i-1,r} - \sum_{r=1}^{s-1} d_{ir} \ = \begin{cases} 1 & \text{if } s : t_{i-1} \leq T_s \leq t_i \\ 0 & \text{else,} \end{cases}$$

which follows from (5.1). Thus, we can write (2.5) as

$$F \leq a_i \sum_{j=1}^{|\mathcal{M}|} z_{ij} \sigma_{sj} + (1 - p_{is}) F_{\max} \quad \forall s = 1, \ldots, D-1 \quad \forall i = 1, \ldots, N+1, \tag{5.8}$$

where $F_{\max}$ is an upper bound on F.

We can further tighten the formulation by modeling the implications

$$\begin{aligned} d_{ir} = 1 \Rightarrow \sum_{s=r}^{D} d_{j,s} = 0 \quad \forall j = 1 \ldots i-1,\ i = 1 \ldots N+1,\ r = 1 \ldots D \\ d_{ir} = 1 \Rightarrow \sum_{s=1}^{r} d_{j,s} = 0 \quad \forall j = i+1 \ldots N+1,\ i = 1 \ldots N+1,\ r = 1 \ldots D, \end{aligned}$$

which is modeled as the valid inequalities

$$\sum_{j=1}^{i-1} \sum_{s=r}^{D} d_{js} + \sum_{j=i+1}^{N+1} \sum_{s=1}^{r} d_{j,s} \leq B(1 - d_{ir}) \quad \forall i = 1 \ldots N+1,\ r = 1 \ldots D, \tag{5.9}$$

where $B \equiv 2N$ is an upper bound on the left-hand side of the inequality.

The resulting model is a smooth MINLP, to which standard MINLP solution techniques can now be applied to find a local solution.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{N+1} q_i \left(\sum_{r=1}^{D} (d_{i-1,r} - d_{ir}) \hat{C}_r \left(\frac{T_H}{T_r} - 1 \right) \right) \; := \; CoolingPower \\
\text{subject to} \quad & (3.13), (3.14), (3.16), (3.25), (3.32), (3.28), (3.33) \\
& (3.38), (4.9), (4.10), (4.11) \\
& (5.1), (5.2), (5.3) \\
& (5.6), (5.7), (5.8), (5.9) \qquad \text{(P-2)} \\
& t_0 = T_C \quad \text{and} \quad t_{N+1} = T_H \\
& y_i \in \{0,1\}, \quad \forall i = 1, \ldots, N+1 \\
& z_{ij} \in \{0,1\}, \quad \forall i = 1, \ldots, N+1,\; j = 1, \ldots, |\mathcal{M}|, \\
& d_{ir} \in \{0,1\}, \quad \forall i = 1, \ldots, N+1,\; r = 1, \ldots, D, \\
& x_i, q_i, u_i, v_{ij}, w_{ij}, n, t_i, a_i \in \mathbb{R}
\end{aligned}$$

Model (P-2) has fewer nonlinear constraints than either (P-0) or (P-1) and is a smooth MINLP. On the other hand, the additional SOS-1 variables d_{ir} make the model larger.

6 Solution Methodology and Computational Results

We have experimented with three models with different smoothness properties. All models are available in AMPL (Fourer et al., 2003) from the authors upon request. The main characteristics of the three models for $N = 10$ are summarized in Table 4. The problem size is roughly linear in N, and the model with $N = 20$ has about twice as many variables and constraints as the model with $N = 10$.

Table 4: Comparison of MINLP Models for $N = 10$

	# Variables	# Constraints	
Model	(bin/SOS/int/cont)	(nonlinear/linear)	Smoothness
MVP			discontinuous objective & constraints
(P-0)	144 (88/0/1/45)	212 (115/97)	discontinuous objective
(P-1)	174 (118/0/1/55)	262 (115/147)	nonsmooth constraints
(P-2)	3696 (3596/11/1/99)	6715 (3281/3434)	smooth as a whistle

We employ standard solution techniques for the solution of the MINLP models (P-0), (P-1), and (P-2) such as branch-and-bound method (Dakin, 1965; Gupta and Ravindran, 1985), and the LP/NLP-based branch-and-bound method (Quesada and Grossmann, 1992); see Grossmann (2002) for a recent survey of solution techniques for MINLP problems.

We note that the models (P-0), (P-1), and (P-2) are nonconvex, making it hard to solve these models to global optimality. We employ a standard nonlinear branch-and-bound-based solver, MINLP-BB (Leyffer, 1998), for solving these models. The main idea behind branch-and-bound is to solve continuous relaxations of the original problem and to divide the feasible region, eliminating the fractional solution of relaxed problem. Continuing in this manner yields a tree of problems that is searched for the integer optimum. The NLP subproblem solved at a node provides a valid lower bound for the subproblems in the descendant nodes. An integer feasible solution when found provides an upper bound for the problem.

We resort to the strategy of multistarts for solving the nonconvex models (P-0) and (P-1). The strategy of multistarts involves solving a problem at a node a number of times from various starting points. For every node of the branch-and-bound tree, we store a logical switch that indicates whether to perform multiple starts. Initially, this switch is set to true, and we perform $R(= 5)$ restarts from randomly generated starting points and select the best solution value obtained as the solution of that node. If all runs give the same solution value, then we set the logical switch to false for this node and for all its children. Otherwise, the switch remains true for all child nodes. Thus, for convex problems, we perform a multistart only at the root node.

We also set priorities on the integer variables in the model for MINLP-BB to perform branching. For the model (P-0), the integer variable n is given the highest priority, so that the solver branches on a fractional value of the variable n, before branching on other variables. The variables y_i have the next highest priority, since they are dependent on n. The variables for the choice of materials, z_{ik} are accorded the next highest priority, since they are dependent on both n and y_i. For the model (P-1), the variables for the discontinuous objective coefficients, s_{ik} are given the least priority.

We ran these models on a Beowulf cluster of computers consisting of 120 nodes of 64-bit AMD Opteron microprocessors. Each of the nodes has a CPU clockspeed of 1.8 GHz, has 2 GB RAM, and runs on the Fedore Core 2 operating system. The models were run without any time limits, with the aim of letting the branch-and-bound enumeration run to completion. For the model (P-0), we did two different runs, choosing the parameter N, the maximum number of intercepts in the system, to be 10 and 20. The results of the runs for $N = 10$ and $N = 20$ are summarized in Table 5. By increasing the number of intercepts, we are able to improve the solution obtained by Abramson (2004) from 1.0623 to 1.039 and 0.988 for $N = 10$ and $N = 20$, respectively.

Table 5: Results for runs for model (P-0)

	Results for $N = 10$				Results for $N = 20$			
	Number of intercepts n: 10				Number of intercepts n: 20			
	Objective solution value f^*: 1.039				Objective solution value f^*: 0.988			
	Number of nodes : 3979				Number of nodes : 13515			
	Number of QPs solved : 1559347				Number of QPs solved : 6229783			
	Time taken (in seconds) : 29894.38				Time taken (in seconds) : 283034.29			
i	t_i	x_i	a_i	z_{ik}	t_i	x_i	a_i	z_{ik}
0	4.2	.	.	.	4.2	.	.	.
1	4.21	1.00	3.102	EPOXYP	4.21	1.00	3.102	EPOXYN
2	7.78	8.01	3.102	EPOXYN	5.78	3.99	3.102	EPOXYN
3	13.58	9.45	3.101	EPOXYN	7.92	4.29	3.102	EPOXYN
4	22.69	11.15	3.100	EPOXYN	10.53	4.74	3.101	EPOXYN
5	39.24	13.99	3.093	EPOXYN	13.91	5.17	3.101	EPOXYN
6	70.88	16.01	3.029	EPOXYN	18.19	5.54	3.100	EPOXYN
7	71	1.05	2.913	EPOXYP	23.68	6.23	3.099	EPOXYN
8	122.21	13.003	3.101	EPOXYN	31.34	7.29	3.091	EPOXYN
9	174.15	8.807	3.734	EPOXYN	41.05	7.17	3.060	EPOXYN
10	179.26	1.00	3.809	EPOXYP	54.66	7.97	3.022	EPOXYN
11	300	16.522	4.474	EPOXYN	70.99	7.66	2.964	EPOXYN
12	.	.	.	.	71	1.00	2.912	EPOXYP
13	.	.	.	.	91.63	5.42	2.928	EPOXYN
14	.	.	.	.	120.99	5.95	3.091	EPOXYN
15	.	.	.	.	146.8	4.54	3.363	EPOXYN
16	.	.	.	.	173.31	4.40	3.721	EPOXYN
17	.	.	.	.	204.29	4.86	4.133	EPOXYN
18	.	.	.	.	228.38	3.57	4.340	EPOXYN
19	.	.	.	.	261.37	4.50	4.472	EPOXYN
20	.	.	.	.	291.3	3.61	4.478	EPOXYN
21	.	.	.	.	300	1.00	4.478	EPOXYN

We note that solving the model with $N = 20$ intercepts is computationally intensive. In order to solve these models faster, we solve the models with the number of intercepts n fixed at $n = 1, \ldots, 29$. This allows us to also fix the indicator variables y_i, reducing the number of variables and the solution time significantly. Figure 5 shows the value of objective function f as a function of the number of intercepts, n. We see that the objective function value first drops sharply as n is increased and then stablizes as the value of n becomes larger. Clearly, $n = 10$ is not the optimal number of intercepts.

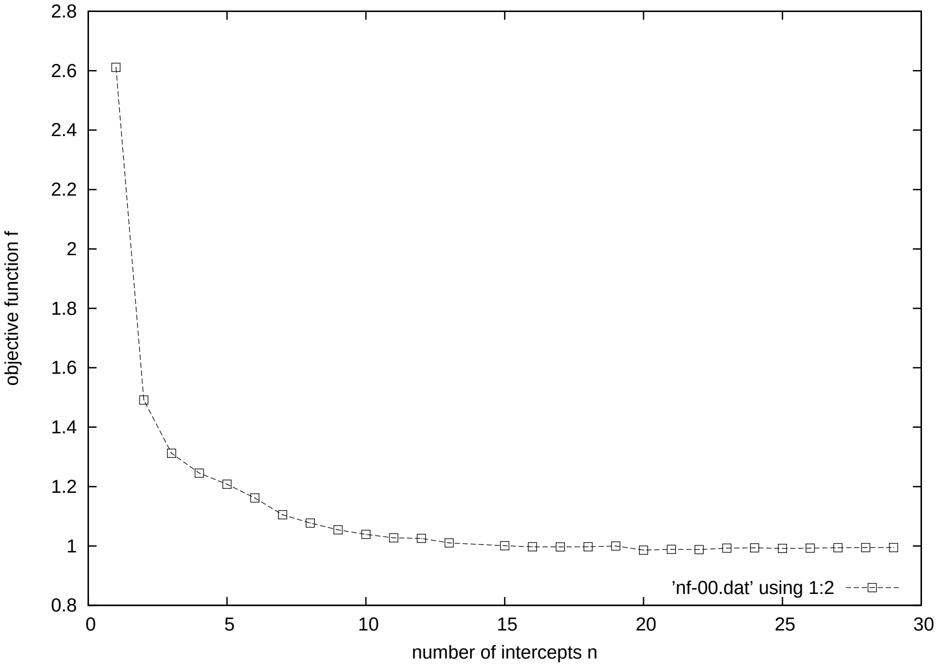

Figure 5: Plot showing objective function value for (P-0) with n.

We performed a similar set of experiments for the model P-1. The results of the runs for $N = 10$ and the best solution found by our runs are summarized in Table 6. The solution obtained by solving the MINLP models (P-0) and (P-1) are similar to or better than the ones obtained by Abramson (2004). Also, by increasing the number of intercepts in the model, we are able to obtain better solutions than the model with at most 10 intercepts. The MINLP model formulated makes it possible to search in the larger sub-space of all the possible materials at an intercept. Thus, we are able to obtain solutions where different materials are used as insulators in different layers in the same configuration. We see that the material chosen in the final configuration is either epoxy(normal) or epoxy(plane).

The addition of new binary variables s_{ki} to model the discontinuous objective function coefficients makes the NLP easier to solve in terms of the number of QP's that are needed to solve per NLP. Table 7 shows the average number of QP's solved per NLP for the models (P-0) and (P-1). We observe that this

Table 6: Results for runs for model (P-1)

	Results for $N = 10$				Best Solution Found for (P-1)			
	Number of intercepts n: 10				Number of intercepts n: 21			
	Objective solution value f^*: 1.02165				Objective solution value f^*: 0.98305			
	Number of nodes : 382540				Number of nodes : 244985			
	Number of QPs solved : 7049146				Number of QPs solved : 9360693			
	Time taken (in seconds) : 28778.95				Time taken (in seconds) : 74283.38			
i	t_i	x_i	a_i	z_{ik}	t_i	x_i	a_i	z_{ik}
0	4.2	.	.	.	4.2	.	.	.
1	4.21	1.00	3.102	EPOXYN	4.21	1.00	3.102	EPOXYN
2	7.78	8.19	3.102	EPOXYN	5.71	3.75	3.102	EPOXYN
3	13.58	9.66	3.101	EPOXYN	7.61	4.07	3.102	EPOXYN
4	22.66	11.41	3.100	EPOXYN	10.03	4.41	3.101	EPOXYN
5	39.17	14.32	3.093	EPOXYN	13.05	4.78	3.101	EPOXYN
6	70.99	16.49	3.029	EPOXYN	16.79	5.17	3.101	EPOXYN
7	71	1.00	2.913	EPOXYP	21.43	5.62	3.000	EPOXYN
8	112.92	9.73	3.034	EPOXYN	27.41	6.19	3.096	EPOXYN
9	154.38	7.37	3.461	EPOXYN	34.57	6.13	3.076	EPOXYN
10	201.53	7.66	4.108	EPOXYP	43.12	6.09	3.048	EPOXYN
11	300	13.16	4.474	EPOXYN	55.46	7.06	3.014	EPOXYN
12	.	.	.	.	70.99	6.78	2.961	EPOXYN
13	.	.	.	.	71	1.00	2.912	EPOXYN
14	.	.	.	.	91.10	5.21	2.927	EPOXYN
15	.	.	.	.	109.95	3.96	3.013	EPOXYN
16	.	.	.	.	128.48	3.44	3.162	EPOXYN
17	.	.	.	.	147.06	3.21	3.367	EPOXYN
18	.	.	.	.	165.84	3.12	3.616	EPOXYN
19	.	.	.	.	184.96	3.09	3.890	EPOXYN
20	.	.	.	.	206.75	3.40	4.157	EPOXYN
21	.	.	.	.	233.52	3.95	4.371	EPOXYN
22	.	.	.	.	300	8.50	4.475	EPOXYN

number remains almost constant for (P-1), while it increases in a near-linear fashion for (P-0). For the largest models we observe an order of magnitude reduction in the number of QP's solved per NLP. We believe that this effect is due to the lack of continuity in model (P-0).

Table 7: Results showing the average number of QP's solved per NLP for (P-0) and (P-1)

n	(P-0)	(P-1)	n	(P-0)	(P-1)
1	26.6	25.15	9	388.36	29.34
2	49.77	33.20	10	407.48	30.26
3	124.77	34.70	19	512.43	32.30
4	209.97	42.25	21	522.03	38.20
5	247.01	35.16	23	528.14	41.05
6	300.07	33.21	25	532.64	43.32
7	300.15	29.32	28	544.19	47.26
8	342.64	32.23	.	.	.

The discretized model (P-2) is smooth as a result of the modeling of temperature at intercepts using discrete variables. Even though the nonlinearity in the model has been considerably reduced, it is still nonconvex. It also has a large number of variables and constraints. We attempted to solve this model using MINLP-BB but observed excessive solution times. Hence, we report the solution statistics only for (P-2) with FilMINT (Abhishek et al., 2006), a linearizations-based solver for mixed-integer nonlinear programs. FilMINT uses MINTO (Nemhauser et al., 1994) for its MILP branch-and-cut framework, and filterSQP (Fletcher et al., 2002), an active-set solver for solving nonlinear programs. It implements the LP/NLP algorithm in a branch-and-cut framework. We refer the reader to Quesada and Grossmann (1992) and Abhishek et al. (2006) for the details of the algorithm. For nonconvex problems, the solution methodology can be seen essentially as a heuristic. FilMINT advanced MILP features make it possible to obtain good upper bounds quickly which help the solution scheme for the problem. For the nonconvex model (P-2), we use the basic version of the LP/NLP algorithm, because linearizations added to the model may cut off the feasible region.

We note that the full 1-degree discretization that we do for model (P-2) is not necessary, because we can be reasonably sure that the intercepts close to the cold end will not have high temperatures, and vice versa. We reduce the discretization level of temperature for an intercept i by restricting the intercept to lie in a smaller interval $[T_{l_i}, T_{u_i}]$ than from $[T_1, T_D]$, where $l_i, u_i \in [1, D]$. This implies modeling the SOS-1 variables d_{ir} as

$$t_i = \sum_{r=l_i}^{u_i} d_{ir} T_r, \quad 1 = \sum_{r=l_i}^{u_i} d_{ir}, \quad d_{ir} \in \{0,1\}, \quad \forall i = 0, \ldots, N+1. \tag{6.1}$$

This reduces the number of integer variables in the model and makes the problem easier to solve. This procedure of choosing different temperature intervals $[T_{l_i}, T_{u_i}]$ is essentially a heuristic with great flexibility. We base our discretization reduction strategy around the solution of models (P-0) and (P-1). For $n = 10$, Table 8 shows the temperature intervals that we allow for different intercepts as part of our strategy.

Table 8: Table showing the discretization reduction strategy for (P-2) for $n = 10$

Intercept i	Temperature Range	Intercept i	Temperature Range
1	$[1, 81]$	6	$[55, 135]$
2	$[1, 81]$	7	$[55, 135]$
3	$[1, 81]$	8	$[82, 189]$
4	$[1, 81]$	9	$[82, 189]$
5	$[1, 81]$	10	$[163, 243]$

We also restrict the choice of materials in the model to epoxy-plane and epoxy-normal. This is a reasonable assumption because we see that the solution obtained for models (P-0) and (P-1) have only these materials in their final configuration. To better handle the nonconvex constraints, we employ the strategy of adding linearizations as local cuts in some of our runs. Hence, the cuts generated at some node are added only to the subtree rooted at that node, instead of the default FilMINT strategy to adding these linearizations globally on all open nodes. For nonconvex problems, this heuristic helps in reducing the feasible region that can be cut off. The results of the runs for $N = 10$ with the mentioned discretization reduction strategy and of the runs using local cuts are shown in Table 9.

We note that the solution obtained in our runs is only a little worse compared to models (P-0) and (P-1). We see that local cuts help in reducing, to a large extent, the feasible region from being cut off, thus letting the branch-and-cut enumeration visit more nodes and find better feasible solutions. We also see that the average number of QP's is still small for the smooth model (P-2) compared to the size of the model. We use the solution of the model (P-2) to obtain the value of the discrete variables n, and $m_i \in \mathcal{M}$, and we rerun an NLP to adjust the values of the structural variables. The results are given in Table 10.

Table 9: Results for runs for model (P-2) for $N = 10$ using FilMINT

	Best Solution for (P-2) with Global Cuts				Best Solution for (P-2) with Local Cuts			
	Number of intercepts n: 10				Number of intercepts n: 10			
	Objective solution value f^*: 1.281264				Objective solution value f^*: 1.212189			
	Number of MILP nodes : 5411				Number of MILP nodes : 15111			
	Number of LP's solved : 6279				Number of LP's solved : 27357			
	Number of NLP's solved : 5				Number of NLP's solved : 7102			
	Number of QP's solved : 158				Number of QP's solved : 286392			
	Average number of QP's per NLP : 31.6				Average number of QP's per NLP : 40.32			
	Time taken (in seconds) : 5712.02				Time taken (in seconds) : 72007.67			
i	t_i	x_i	a_i	z_{ik}	t_i	x_i	a_i	z_{ik}
0	4.2	.	.	.	4.2	.	.	.
1	16	25.21	3.106	EPOXYN	6	5.54	3.103	EPOXYN
2	40	21.40	3.104	EPOXYN	7	1.99	3.104	EPOXYN
3	41	1.00	3.029	EPOXYP	8	1.85	3.105	EPOXYN
4	71	15.95	3.025	EPOXYN	9	1.73	3.105	EPOXYN
5	72	1.00	2.908	EPOXYN	15	8.91	3.106	EPOXYN
6	137	12.59	3.248	EPOXYN	67	40.56	3.105	EPOXYN
7	138	1.00	3.259	EPOXYN	72	3.86	2.917	EPOXYN
8	191	7.89	3.976	EPOXYN	124	10.75	3.116	EPOXYN
9	192	1.00	3.989	EPOXYN	160	5.73	3.537	EPOXYN
10	204	1.61	4.136	EPOXYN	174	2.49	3.736	EPOXYP
11	300	11.33	4.483	EPOXYN	300	16.58	4.483	EPOXYN

Table 10: Results for model (P-0) after fixing discrete variables using (P-2)

	Results for $n = 10$								
	Objective Solution Value f^*: 1.097219								
	Number of QPs Solved : 508								
i	t_i	x_i	a_i	z_{ik}	i	t_i	x_i	a_i	z_{ik}
0	4.2	.	.	.	6	28.79	8.83	3.099	EPOXYN
1	4.21	1.01	3.102	EPOXYN	7	41.03	8.95	3.071	EPOXYN
2	6.44	5.09	3.106	EPOXYN	8	71	16.23	3.022	EPOXYN
3	9.59	5.70	3.102	EPOXYN	9	134.2	16.33	3.221	EPOXYN
4	13.91	6.38	3.101	EPOXYN	10	155.48	4.02	3.476	EPOXYP
5	19.75	7.13	3.100	EPOXYN	11	300	20.33	4.475	EPOXYN

7 Conclusions

We use mixed integer nonlinear programming techniques to model the load-bearing thermal insulation problem. We use integer variables to model the categorical variables, so that the model now allows continuous relaxations and can be solved by using standard MINLP solution techniques. We develop facet-defining inequalities for a relaxation of this reformulated MINLP model. We evaluate integrals by adding more data points consistent with the cubic interpolation of the data and using trapezoidal rule. We also avoid the second-level optimization in the problem by introducing a finer piecewise linear approximation of the data and by enforcing the bounds at temperatures for each intercept.

Our reformulations give rise to three models with varying degrees of smoothness, and we comment on the relative merits of the formulations. Our computational results indicate that the MINLP formulations obtain better results than previous results by Abramson (2004). In particular, by increasing the number of intercepts from 10 to 20, we are able to reduce the cooling power by 4%.

The modeling of mixed variable problems as MINLPs allows us to apply more powerful techniques, such as branch-and-bound, outer approximation, and branch-and-cut, rather than a heuristic search technique. Engineering or modeling insight is included into the MILP model by using priorities on the integer variables or by restricting temperature ranges for the intercepts. We believe that the modeling techniques shown here are very general and can be used as a blueprint for modeling other design problems that have categorical variables.

Acknowledgments

Much of this work was carried out while the first author was visiting Argonne through a student visitor program, made possible through the support by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. This work was also supported by the U.S. Department of Energy through the grant DE-FG02-05ER25694.

References

Abhishek, K., Leyffer, S., and Linderoth, J. (2006). FilMINT: An outer approximation based nonlinear mixed integer solver. Working Paper.

Abramson, M. A. (2004). Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm. *Optimization and Engineering*, 5:157–177.

Abramson, M. A., Audet, C., and J. E. Dennis, J. (2004). Filter pattern search algorithms for mixed variable constrained optimization problems. Technical Report TR04-09, CAAM, Rice University.

Audet, C. and Dennis, J. (2004). A pattern search filter method for nonlinear programming without derivatives. *SIAM J. Optimization*, 14(4):980–1010.

Audet, C. and Dennis, J. E. (2000). Pattern search algorithms for mixed variable programming. *SIAM J. on Optimization*, 11(3):573–594.

Beal, J. M., Shukla, A., Brezhneva, O. A., and Abramson, M. A. (2006). Optimal sensor placement for enhancing sensitivity to change in stiffness for structural health monitoring. Technical report, Department of Mathematics and Statistics, Air Force Institute of Technology.

Dakin, R. J. (1965). A tree search algorithm for mixed programming problems. *Computer J.*, 8:250–255.

Fletcher, R., Leyffer, S., and Toint, P. (2002). On the global convergence of a Filter-SQP algorithm. *SIAM J. Optimization*, 13:44–59.

Fourer, R., Gay, D. M., and Kernighan, B. W. (2003). *AMPL: A Modelling Language for Mathematical Programming*. Books/Cole—Thomson Learning, 2nd edition.

Grossmann, I. E. (2002). Review of nonlinear mixed–integer and disjunctive programming techniques. *Optimization and Engineering*, 3:227–252.

Gupta, O. K. and Ravindran, A. (1985). Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31:1533–1546.

Kokkolaras, M., Audet, C., and Dennis, J. E. (2001). Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system. *Optimization and Engineering*, 2:5–29.

Leyffer, S. (1998). User manual for MINLP-BB. University of Dundee.

Nemhauser, G. and Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. John Wiley and Sons, New York.

Nemhauser, G. L., Savelsbergh, M. W. P., and Sigismondi, G. C. (1994). MINTO, a Mixed INTeger Optimizer. *Operations Research Letters*, 15:47–58.

Quesada, I. and Grossmann, I. E. (1992). An LP/NLP based branch–and–bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947.

Zhao, Z., Meza, J., and van Hove, M. (2005). Using pattern search methods for surface structure determination of nanostructures. Technical Report LBNL-57541, Lawrence Berkeley National Laboratory.