Applications and algorithms for mixed integer nonlinear programming

Sven Leyffer[1], **Jeff Linderoth**[2], **James Luedtke**[2], **Andrew Miller**[3], **Todd Munson**[1]

[1] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

[2] Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA

[3] IMB, Université Bordeaux 1; RealOpt, INRIA Bordeaux Sud-Ouest, 33405 Talence, France

E-mail: `linderoth@wisc.edu`

Abstract. The mathematical modeling of systems often requires the use of both nonlinear and discrete components. Discrete decision variables model dichotomies, discontinuities, and general logical relationships. Nonlinear functions are required to accurately represent physical properties such as pressure, stress, temperature, and equilibrium. Problems involving both discrete variables and nonlinear constraint functions are known as *mixed-integer nonlinear programs* (MINLPs) and are among the most challenging computational optimization problems faced by researchers and practitioners. In this paper, we describe relevant scientific applications that are naturally modeled as MINLPs, we provide an overview of available algorithms and software, and we describe ongoing methodological advances for solving MINLPs. These algorithmic advances are making increasingly larger instances of this important family of problems tractable.

1. Introduction

A mixed-integer nonlinear program (MINLP) is the numerical optimization problem of finding

$$z_{\text{MINLP}} = \underset{x \in X, y \in Y \cap \mathbb{Z}^p}{\text{minimize}} \; f(x,y) \;\; \text{subject to } g(x,y) \leq 0, \qquad \text{(MINLP)}$$

where $f : \mathbb{R}^{n+p} \to \mathbb{R}$ and $g : \mathbb{R}^{n+p} \to \mathbb{R}^m$ are continuously differentiable functions; x and y are continuous and discrete variables, respectively; and X and Y are compact polyhedral subsets of $\mathbb{R}^n$ and $\mathbb{R}^p$, respectively. If the objective function $f(x,y)$ and the constraint function $g(x,y)$ are convex functions, then the problem is known as a *convex MINLP*, otherwise the problem is a *nonconvex MINLP*.

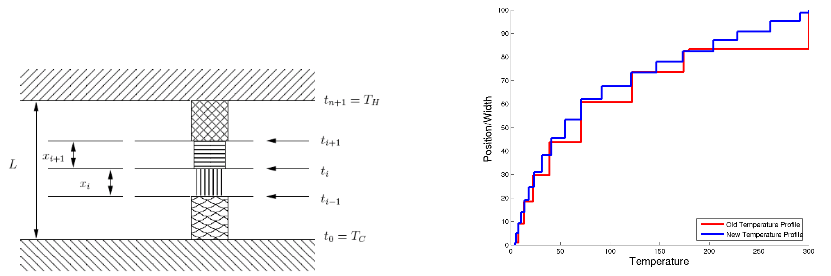
Given their generality and flexibility, MINLPs have been proposed for many diverse and important scientific applications, including the efficient management of electricity transmission [1], contingency analysis and blackout prevention of electric power systems [2], the design of water distribution networks [3], operational reloading of nuclear reactors [4], and minimization of the environmental impact of utility plants [5].

Unfortunately, current algorithms and available software are often unable to solve practically-sized instances of these important models. Our research is aimed at correcting the mismatch between natural optimization models and available robust optimization solvers. We hope to enable application scientists to confidently employ MINLP tools to address their challenging scientific problems.

2. Applications

Design of thermal insulation layer for the large hadron collider: A thermal insulation system uses a series of heat intercepts and surrounding insulators to minimize the power required to maintain the heat intercepts at certain temperatures. Problems of this type arise in the design of superconducting magnetic energy storage systems and have been used in the Large Hadron Collider project. Figure 1-(a) shows the conceptual design.

The designer chooses the maximum number of intercepts, the thickness and area of each intercept, and the material of intercept from a discrete set of materials. The choice of material affects the thermal conductivity and total mass of the insulation system. Nonlinear functions in the model are required to accurately model system characteristics such as heat flow between the intercepts, thermal expansion, and stress constraints. Integer variables are used to model the discrete choice of the type of material to use in each layer.



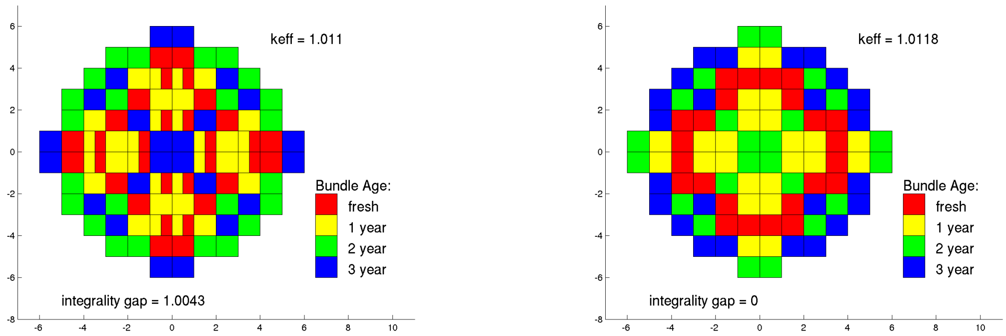
(a) Conceptual Design of Insulation Layer

(b) Optimal Temperature Profile

Figure 1. Design of Thermal Insulation

In [6], by using MINLP, we have identified solutions that were as much as 4% better than those identified by any previously available method. The improvement is largely due to the ability to handle a larger number of intercepts than previously possible. The optimal temperature profile of the new design is shown as the blue curve in Figure 1-(b), compared to the red curve for the previous suboptimal design. The area between the two profiles corresponds to the savings in cooling power.

Nuclear core-reload: The fuel elements in a nuclear reactor core are divided into groups of different ages. At the end of each cycle the oldest group is removed, the remaining groups are reshuffled, and a fresh group is brought in. Finding a pattern for how to move the elements to maximize fuel efficiency, subject to power safety constraints, leads to a mixed-integer nonlinear problem. The model includes dependent variables that describe physical properties such as neutron flux, burn-up, and yield. The neutron transport equations are converted to a set of algebraic equations using Green's functional theory, giving rise to a stationary description of the neutron flux in the core. The fuel burn-up is approximated by discretizing the differential equation. The resulting MINLP contains binary variables that model the reloading operation and algebraic equations arising from the discretization of the PDE. Figure 2-(a) shows an initial design that ignores the integrality of the reloading operation, and Figure 2-(b) shows the optimal reloading pattern, where different colors represent the different age groups. It is important to note that simple rounding of the solution in Figure 2-(a) does not produce a valid reload-pattern.



(a) Solution to NLP Relaxation

(b) Solution to MINLP

Figure 2. Core-reload Problem Solutions

3. Algorithms for MINLP

The basic operations common to algorithms for solving (MINLP) are *relax* and *search*. Algorithms differ in the manner these operations are performed. For convex MINLPs, the most natural relaxation is to disregard the integrality constraints $y \in \mathbb{Z}^p$. The resulting nonlinear program (NLP), when solved to global optimality, provides a lower bound on the optimal solution value z_{MINLP}. If the solution $(\hat{x}, \hat{y})$ to the relaxation has $y \in \mathbb{Z}^p$, then $(\hat{x}, \hat{y})$ must be the optimal solution to (MINLP), otherwise there is a $j \in \{1, \ldots, p\}$ with $\hat{y}_j \notin \mathbb{Z}$. In *branch-and-bound*, the search continues by creating two new subproblems, one with the additional constraint $y_j \leq \lfloor \hat{y}_j \rfloor$, and one with the additional constraint $y_j \geq \lceil \hat{y}_j \rceil$. Each of these two subproblems is then (recursively) solved by the same procedure, resulting in a search-tree of subproblems that are evaluated. Software that implements a branch-and-bound algorithm for convex MINLPs is listed in Table 1. The software packages differ in the way the search-tree is created and in the NLP algorithm used to solve relaxations.

Unfortunately, if $f(x, y)$ or $g(x, y)$ are not convex functions, then standard algorithms for solving NLP are able to guarantee convergence only to a local minimum. Thus, the solution of the relaxation *does not* provide a lower bound on z_{MINLP}. In this case of non-convex MINLP, an additional relaxation step is required. Typically, the relaxation is based on a decomposition of the nonlinear functions into components, and for each component, a "convex envelope" relaxation is built [7]. Branch and bound is again applied to the relaxation, with the additional complication that continuous variables x may require branching to improve the convex relaxation of the original nonconvex functions. Solvers of this class (listed in Table 1) differ in the manner in which relaxations are created. The impact of creating strong relaxations for nonconvex functions is briefly demonstrated in Section 4.2.

The final class of algorithms developed for solving convex MINLP are those based on linearizations. If $f(x, y)$ and $g(x, y)$ are convex, then for any point $(\hat{x}, \hat{y})$ the inequalities

$$f(\hat{x}, \hat{y}) + \nabla f(\hat{x}, \hat{y})^T \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \leq f(x, y) \quad \text{and} \quad g(\hat{x}, \hat{y}) + \nabla g(\hat{x}, \hat{y})^T \begin{bmatrix} x - \hat{x} \\ y - \hat{y} \end{bmatrix} \leq 0, \qquad (1)$$

underestimate the objective function and form a linear outer approximation of the feasible region, respectively. In linearization-based methods, a linear approximation to (MINLP) is dynamically created by applying the inequalities (1) at many different points $(\hat{x}, \hat{y})$. The approximation is solved by a branch-and-bound procedure to ensure convergence. There is significant flexibility in where linearizations are taken and how they are combined with the branch-and-bound search. Software packages with implementations of linearization-based algorithms are listed in Table 1. Most of the codes listed in Table 1 are available for use free of charge via the NEOS server: `http://www-neos.mcs.anl.gov/`.

Table 1. Software for solving MINLP

Algorithm	**Software**
Branch and Bound, Convex	Bonmin, KNITRO, MINLP-BB, SBB
Branch and Bound, Nonconvex	BARON, Couenne, LINDO-Global
Linearization-Based, Convex	Alpha-ECP, Bonmin, Dicopt, FilMINT

4. Strong reformulations

Recognizing and exploiting instance-specific structure in (MINLP) to create strong reformulations can have a huge impact on an algorithm's ability to solve an instance. We conclude with two brief examples demonstrating this point.

4.1. Strong reformulations for convex MINLPs

Integer variables in MINLPs are often used as "indicator variables." For example, consider

$$S = \{(x, y, z) \in \mathbb{R}^2 \times \{0, 1\} \ : \ y \geq x^2, \ \ uz \geq x \geq lz, \ x \geq 0\}, \text{ where } u, l \in \mathbb{R}. \tag{2}$$

Setting the indicator variable $z = 0$ forces $x = 0$, but when $z = 1$, the inequalities $x \in [l, u], y \geq x^2$ are enforced. We have shown in [8] that the convex hull of S is given by the *perspective reformulation*: $\text{conv}(S) = \left\{(x, y, z) \in \mathbb{R}^3 \ : \ yz \geq x^2, \ uz \geq x \geq lz, \ 1 \geq z \geq 0, \ x, y \geq 0\right\}$. Reformulating simple sets S that appear in (MINLP) in the form $\text{conv}(S)$ will significantly strengthen the relaxation and improve algorithm performance.

As a concrete example, consider a Separable Quadratic Uncapacitated Facility Location Problem (SQUFL) [9]. The MINLP formulation of SQUFL has constraints of the form

$$x_{ij}^2 - y_{ij} \leq 0 \quad \text{ and } \quad y_{ij} \leq z_i \quad \forall i, j. \tag{3}$$

The indicator variable z_i should take the value 1 if facility i is opened. If $z_i = 0$, then the constraints (3) force $y_{ij} = x_{ij} = 0$, meaning that no demand for customer j can be served from facility i. Since inequalities (3) are of the form (2), the perspective reformulation may be applied, replacing $x_{ij}^2 - y_{ij} \leq 0$ with $x_{ij}^2 z_i - y_{ij} \leq 0 \ \forall i, j$.

In [10], an instance with 30 facilities and 100 customers was solved by applying Bonmin to the original reformulation. The solution required 16,697 CPU seconds and 45,901 nodes in the search-tree. By applying the perspective reformulation, the same instance was solved in *23 CPU seconds*, enumerating 44 nodes of the search tree [8]. The speedup factor is more than 700.

4.2. Strong relaxations of multilinear forms

A particular nonconvex structure we are studying is the case of *multilinear* functions of the form $\phi(x) = \sum_{i=1}^m a_i \prod_{j \in S_i} x_j$ where $a_i \in \mathbb{R}$ and $S_i \subseteq N := \{1, \ldots, n\}$ and we assume $l \leq x \leq u$ for some $l, u \in \mathbb{R}^n$. Many applied problems have terms of this form, either explicitly (e.g. the pooling problem [11]) or after reformulation. The "textbook" approach for obtaining relaxations of such problems is to reformulate the problem (by adding new variables) into a problem in which the only nonconvexities remaining are bilinear constraints of the form $x_k = x_i x_j$, and then to relax each of these constraints using a set of simple linear inequalities known as the the McCormick inequalities [12]. We consider an alternative approach in which we use the convex hull of the set $\{z \in \mathbb{R}, x \in \mathbb{R}^n \mid z = \phi(x), \ l \leq x \leq u\}$, which can be obtained by writing the vector x as a convex combination of the vertices of the hypercube containing x [13].

To demonstrate the impact of this reformulation technique, we conducted a preliminary test on the following example problem:

$$\underset{l \leq x \leq u}{\text{minimize}} \ \ 4(x_1 + x_2 + x_3) + 3(x_4 + x_5) + 3.5(x_6 + x_7) + 2.5x_8$$

$$\text{subject to } x_1x_2x_3x_4 - x_1x_2 - x_4x_5 + x_5 + x_6 \geq 230, \ \ x_3x_4x_5x_6 - x_1x_4 - x_6x_7 + x_2 + x_8 = -2,$$

where $l = [-1, -2.5, -0.5, -0.5, 1, 1, 1, 1]$ and $u = [4, 4, 6, 6, 6, 5, 3]$. For this example, the McCormick relaxation based on a bilinear reformulation yields a lower bound of 2.33 on the value of the best possible solution, compared to a significantly better lower bound of 36.33 obtained by solving the convex hull relaxation. We then passed the solution of each of these relaxations to the NLP solver IPOPT [14] in an attempt to find a good feasible solution. IPOPT failed to find a feasible solution when given the McCormick relaxation solution as an initial point, but found a feasible solution of value 72.1 when starting from the convex hull relaxation solution. Solving this problem with the global optimization solver BARON [7] confirmed that this solution is optimal. This example suggests that the convex hull approach may be useful both for yielding stronger lower bounds and for finding better feasible solutions.

Acknowledgments
This work was supported in part by National Science Foundation Grant CCF-0830153 and by the Office of Advanced Scientific Computing Research, Office of Science, U.S Department of Energy, under Contract DE-AC02-06CH11357 and Grants DE-FG02-08ER25861 and DE-FG02-09ER25869.

References

[1] Momoh J, Koessler R, Bond M, Stott B, Sun D, Papalexopoulos A and Ristanovic P 1997 *IEEE Transaction on Power Systems* **12**(1) 444–455

[2] Donde V, Lopez V, Lesieutre B, Pinar A, Yang C and Meza J 2005 Identification of severe multiple contingencies in electric power networks *Proceedings 37th North American Power Symposium* lBNL-57994

[3] Bragalli C, D'Ambrosio C, Lee J, Lodi A and Toth P 2006 An MINLP solution method for a water network problem *Algorithms - ESA 2006 (14th Annual European Symposium. Zurich, Switzerland, September 2006, Proceedings)* (Springer) pp 696–707

[4] Quist A J, van Gemeert R, Hoogenboom J E, Ílles T, Roos C and Terlaky T 1998 *Annals of Nuclear Energy* **26** 423–448

[5] Eliceche A M, Corvalán S M and Martínez P 2007 *Computers and Chemical Engineering* **31** 648–656

[6] Abhishek K, Leyffer S and Linderoth J T 2007 Modeling without categorical variables: A mixed-integer nonlinear program for the optimization of thermal insulation systems Preprint ANL/MCS-P1434-0607 Argonne National Laboratory Argonne, IL

[7] Sahinidis N V 1996 *Journal of Global Optimization* **8** 201–205

[8] Günlük O and Linderoth J 2008 Perspective relaxation of mixed integer nonlinear programs with indicator variables *IPCO 2008: The Thirteenth Conference on Integer Programming and Combinatorial Optimization* vol 5035 ed Lodi A, Panconesi A and Rinaldi G pp 1–16

[9] Günlük O, Lee J and Weismantel R 2007 MINLP strengthening for separable convex quadratic transportation-cost UFL Tech. Rep. RC24213 (W0703-042) IBM Research Division

[10] Lee J 2007 *IBM Journal of Research & Development* **51** 489–497

[11] Floudas C A and Aggarwal A 1990 *ORSA Journal on Computing* **2** 225–235

[12] McCormick G P 1976 *Mathematical Programming* **10** 147–175

[13] Sherali H 1997 *Acta Mathematica Vietnamica* **22** 245–270

[14] Wächter A and Biegler L T 2006 *Mathematical Programming* **106** 25–27