An Extended Mathematical Programming Framework[☆]

Michael C. Ferris[*,a], Steven P. Dirkse[b], Jan-H. Jagla[c], Alexander Meeraus[b]

[a]Computer Sciences Department, University of Wisconsin, Madison, WI 53706
[b]GAMS Corporation, 1217 Potomac Street, Washington, DC 20007
[c]GAMS Software GmbH, Eupener Str. 135{137, 50933 Cologne, Germany

Abstract

Extended mathematical programs are collections of functions and variables joined together using specific optimization and complementarity primitives. This paper outlines a mechanism to describe such an extended mathematical program by means of annotating the existing relationships within a model to facilitate higher level structure identification. The structures, which often involve constraints on the solution sets of other models or complementarity relationships, can be exploited by modern large scale mathematical programming algorithms for efficient solution. A specific implementation of this framework is outlined that communicates structure from the GAMS modeling system to appropriate solvers in a computationally beneficial manner. Example applications are taken from chemical engineering.

Key words: Modeling, complementarity, nonlinear programming, variational inequalities

[☆]This work is supported in part by Air Force Office of Scientific Research Grant FA9550-07-1-0389, and National Science Foundation Grants DMI-0521953, DMS-0427689 and IIS-0511905. The authors wish to thank Todd Munson and Tom Rutherford for insightful comments on the material presented here.

[*]Corresponding author

Email addresses: ferris@cs.wisc (Michael C. Ferris), sdirkse@gams.com (Steven P. Dirkse), jhjagla@gams.com (Jan-H. Jagla), ameeraus@gams.com (Alexander Meeraus)

1. Introduction

Chemical engineering applications often involve the modeling and solution of classical mathematical programs. Indeed, chemical engineering has been a user of optimization techniques and technology for several decades and many chemical engineers have made significant contributions to the field of optimization, and in several cases have developed software that has had impact far beyond the chemical engineering discipline. Packages such as DICOPT (Viswanathan and Grossmann, 1990), BARON (Tawarmalani and Sahinidis, 2004), ALPHAECP (Westerlund et al., 1998) and IPOPT (Wächter and Biegler, 2006) have influenced the debate on tractability of hard, practical nonlinear optimization problems, often setting the gold standard for the solution of nonconvex, global optimization problems.

Accessing these solvers, and many of the other algorithms that have been developed over the past three decades has been made easier by the advent of modeling languages. A modeling language (Bisschop and Meeraus, 1982; Fourer et al., 1990) provides a natural, convenient way to represent mathematical programs and provides an interface between a given model and multiple different solvers for its solution. The many advantages of using a modeling language are well known. They typically have efficient automatic procedures to handle vast amounts of data, take advantage of the numerous options for solvers and model types, and can quickly generate a large number of models. For this reason, and the fact that they eliminate many errors that occur without automation, modeling languages are heavily used in practical applications. Although we will use GAMS (Brooke et al., 1988), the system we are most familiar with, much of what will be said could as well be applied to other algebra based modeling systems like AIMMS (Bisschop and Entriken, 1993), AMPL (Fourer et al., 1993), MOSEL (Dash Optimization, 2009), MPL (Maximal Software, 2009) and OPL (Van Hentenryck, 1999).

The extended mathematical programming (EMP) framework exists to provide these same benefits for applications (e.g. disjunctive programming) that fall outside the classical framework. A high-level description of these models in an algebraic modeling language, along with tools to automatically create the different realizations or extensions possible, pass them on to the appropriate solvers, and interpret the results in the context of the original model, makes it possible to model more easily, to conduct experiments with formulations otherwise too time-consuming to consider, and to avoid errors that can make results meaningless or worse. We believe that further advances in applications of optimization can be achieved via identification of specific problem structures within a model. The EMP framework provides an extensible way to achieve this.

Some motivating examples from the fields of chemical and biological engineering are apparent. For example, the pioneering work of (Clark and Westerberg, 1990) showed how to model design problems under thermodynamic equilibrium conditions as a bilevel optimization problem: in this setting the process design is optimized (typically for operating cost) subject to constraints that involve another optimization problem minimizing Gibbs free energy, along with more typical constraints involving mass and energy balance. Similar problems arise in the optimization of the production of chemicals or biochemicals in metabolic engineering. Of particular note is the work of Maranas and colleagues (Burgard and Maranas, 2003; Burgard et al., 2003), where a bilevel optimization called OptKnock has been developed to identify the (reaction) deletion strategies that couple the cellular objective (inner problem) to the bioengineering objective (outer problem) of maximizing biomass yield or minimizing metabolic adjustment (MOMA). The problem of parameter estimation (for example in thermodynamic equilibrium) can also be framed in this manner (Mitsos et al., 2008; Raghunathan et al., 2006). Such models contain adjustable parameters which cannot be measured directly, but for which an outer problem can minimize errors in predictions from an inner model that encodes system dynamics and properties. Finally, the work by Grossmann and colleagues on generalized disjunctive programming (Turkay and Grossmann, 1996; Vecchietti and Grossmann, 1999; Vecchietti et al., 2003) involves both nonlinear equations and optimization primitives coupled with pure logic relations; this has been used extensively in the synthesis and design of process networks. Application of these ideas for the reduction of gasoline emissions is given in Furman and Androulakis (2008). Each of these examples falls naturally within our EMP framework.

The purpose of this work is to extend the classical nonlinear program from the traditional model:

$$\min_x f(x) \text{ s.t. } g(x) \leq 0,\ h(x) = 0, \tag{1}$$

where f, g and h are assumed sufficiently smooth, to a more general format that allows new constraint types and problem features to be specified precisely. Some extensions of this format have already been incorporated into modeling systems. There is support for integer, semiinteger, and semicontinuous variables, and some limited support for logical constructs including special ordered sets (SOS). GAMS , AMPL and AIMMS have support for complementarity constraints (Ferris and Munson, 2000; Ferris et al., 1999), and there are some extensions that allow the formulation of second-order cone programs within GAMS. AMPL has specific syntax to model piecewise linear functions. Much of this development is tailored to particular constructs within a model. We aim to develop more general annotation schemes to allow extended mathematical programs to be written clearly and succinctly.

In the following sections we outline a new model type that we refer to as an extended mathematical program (EMP). This incorporates many of the extensions mentioned above but also allows a variety of other structures to be described at a model level and is general enough to easily allow further enhancements. We believe such extensions may have benefits on several levels. First, we think this will make the modeler's task easier, in that the model can be described more naturally and perhaps at a higher conceptual level. (Of course, there are several examples of this already in the literature including the use of specialized languages such as MPSGE (Rutherford, 1999) to facilitate general equilibrium models, and specialized (graphical) interfaces to allow queueing system or process system design.) Second, techniques such as automatic differentiation and problem reformulation (duality constructs, or specific ways to reformulate certain constraints) can more reliably do the automatic generation of the model. Third, if an algorithm is given additional structure, it may be able to exploit that in an effective computational manner; knowing the problem is a cone program, or the problem involves the optimality conditions of a nonlinear program can be treated in a variety of different ways, some of which may be distinctly superior to others in certain settings. Indeed, the availability of such structure to a solver may well foster the generation of new features to existing solvers or drive the development of new classes of algorithms. To ensure these extensions are available to a large class of users, we outline within our development a specific implementation within the GAMS modeling system. This prototype is available for general use and will enable both chemical engineers and operations researchers to experiment with new solution strategies based on the provision of extra structural information provided to a solver.

2. Extended Mathematical Programs

The EMP framework allows annotation to existing functions and variables within a model. We begin with the example of complementarity, which in its simplest form, is the relationship between nonnegative variables with the additional constraint that at least one must be zero. In process optimization, complementarity can be used to model (nonsmooth) features such as the disappearance of phases, distillation, flow reversal, safety valve operation, and other discrete events (Gopal and Biegler, 1999; Baumrucker et al., 2008). Following this simple example, we show how the annotations can be incorporated into optimization problems such as mathematical programs with complementarity constraints. This format is used by (Yang et al., 2008) in their optimal (enzymatic) capacity constraint identification (OCCI) method. Such examples lead to natural extensions including variational inequality constraints, bilevel programs and mathematical programs with equilibrium constraints. Further extensions beyond complementarity are given later in the paper.

2.1. Complementarity Problems

The necessary and sufficient optimality conditions for the linear program

$$\begin{aligned} &\min_x c^T x \\ &\text{s.t. } Ax \geq b,\ x \geq 0 \end{aligned} \tag{2}$$

are that x and some λ satisfy the complementarity realtionships:

$$\begin{aligned} 0 \leq c - A^T\lambda \quad &\perp \quad x \geq 0 \\ 0 \leq Ax - b \quad &\perp \quad \lambda \geq 0. \end{aligned} \tag{3}$$

Here, the \$\perp$" sign signifies (for example) that in addition to the constraints $0 \leq Ax - b$ and $\lambda \geq 0$, each of the products $(Ax - b)_i\lambda_i$ is constrained to be zero. An equivalent viewpoint is that either $(Ax - b)_i = 0$ or $\lambda_i = 0$. Within GAMS, these constraints can be modeled simply as

```
positive variables lambda, x;
model complp / defd.x, defp.lambda /;
```

where defp and defd are the equations that define general primal and dual feasibility constraints ($Ax \geq b$, $c \geq A^T\lambda$) respectively.

Other linear programs with specialized constraint structure are just as easy to specify. For example

$$\begin{aligned} &\min_x c^T x \\ &\text{s.t. } Ax = b,\ x \in [l, u] \end{aligned}$$

has similarly expressed optimality conditions:

$$\begin{array}{ll} 0 \leq (c - A^T\lambda)_j & \text{if } x_j = l_j \\ 0 = (c - A^T\lambda)_j & \text{if } l_j < x_j < u_j \\ 0 \geq (c - A^T\lambda)_j & \text{if } x_j = u_j \\ 0 = Ax - b \quad \perp & \lambda \text{ free.} \end{array} \tag{4}$$

Note that the first three complementarity relationships in (4) can be written more succinctly as $(c - A^T\lambda) \perp x \in [l, u]$. This is translated into GAMS syntax as follows:

```
variables lambda, x;
x.lo(i) = l(i); x.up(i) = u(i);
model complp / defd.x, defp.lambda /;
```

Such a problem is an instance of a linear mixed complementarity problem, for which we use the acronym MCP. Note that the bounds on the variables x determine the nature of the relationship on $c - A^T\lambda$ at the solution. (It is possible to introduce explicit multipliers on the constraints $x \geq l$ and $x \leq u$, and to rewrite the optimality conditions in terms of x, λ, and these multipliers. The \$\perp$" notation enables us to write these relationships much more succinctly.)

Complementarity problems do not have to arise as the optimality conditions of a linear program; the optimality conditions of the nonlinear program (1) constitute the following MCP:

$$\begin{array}{lll} 0 = \nabla f(x) + \lambda^T \nabla g(x) + \mu^T \nabla h(x) & \perp & x \text{ free} \\ 0 \leq -g(x) & \perp & \lambda \geq 0 \\ 0 = -h(x) & \perp & \mu \text{ free.} \end{array} \tag{5}$$

Many examples are no longer simply the optimality conditions of an optimization problem. A specific example arises in chemical phase equilibrium. In this setting, different conditions are satisfied at an equilibrium depending on whether we are in vapor, liquid or two-phase state. Letting α represent the fraction in vapor, the problem is to find $f(\alpha) \perp \alpha \in [0,1]$ where

$$f(\alpha) = \sum_{i=1}^{n}(x_i - K_i x_i), \quad x_i = \frac{z_i}{K_i\alpha + 1 - \alpha}, i = 1, \ldots, n$$

for given data K_i and z_i. Gopal and Biegler (1997) have an alternate but equivalent formulation of this model including the multiphase setting. Ferris and Pang (1997) catalogue a number of other applications both in engineering and economics that can be written in a similar format.

It should be noted that robust large scale solvers exist for such problems; see Ferris and Munson (2000) for example, where a description is given of the PATH solver.

2.2. Mathematical Programs with Complementarity Constraints

A mathematical program with complementarity constraints embeds a parametric MCP into the constraint set of a nonlinear program as indicated in the following problem:

$$\min_{x \in \mathbf{R}^n, y \in \mathbf{R}^m} f(x,y) \tag{6}$$

$$\text{s.t. } g(x,y) \leq 0 \tag{7}$$

$$0 \leq y \perp h(x,y) \geq 0. \tag{8}$$

The objective function (6) needs no further description, except to state that the solution techniques we are intending to apply require that f (g and h) are at least once differentiable, and for many modern solvers twice differentiable.

The constraints (7) are intended to represent standard nonlinear programming constraints. Clearly, these could involve equalities with a slight increase in exposition complexity.

The constraints that are of interest here are the complementarity constraints (8). Essentially, these are parametric constraints (parameterized by x) on the variable y, and encode the structure that y is a solution to the nonlinear complementarity problem defined by $h(x, \cdot)$. Within the GAMS modeling system, this can be written simply and directly as:

```
model mpecmod / deff, defg, defh.y /;
option mpec=nlpec;
solve mpecmod using mpec minimizing obj;
```

Here it is assumed that the objective (6) is defined in the equation deff, the general constraints (7) are defined in defg and the function h is described by defh. The complementarity relationship is defined by the bounds on y and the orthogonality relationship shown in the model declaration using \.". AMPL provides a slightly different but equivalent syntax for this, see Ferris et al. (1999). The problem is frequently called a mathematical program with complementarity constraints (MPCC). Several applications of this format within chemical engineering were given above.

Some solvers can process complementarity constraints explicitly. In many cases, this is achieved by a reformulation of the constraints (8) into the classical nonlinear programming form given within (1). GAMS Development Corporation (2008) outline a variety of ways to carry this out, all of which have been encoded in a solver package called NLPEC. Similar strategies are outlined in section 3 of Baumrucker et al. (2008). While there are large numbers of different reformulations possible, the following parametric approach, coupled with the use of the nonlinear programming solver CONOPT or SNOPT, has proven effective in a large number of applications:

$$\begin{aligned} \min_{x\in\mathbf{R}^n, y\in\mathbf{R}^m, s\in\mathbf{R}^m} & f(x,y) \\ \text{s.t. } & g(x,y) \le 0 \\ & s = h(x,y) \\ & y \ge 0, s \ge 0 \\ & y_i s_i \le \mu, \quad i = 1, \ldots, m. \end{aligned}$$

Note that a series of approximate problems are produced, parameterized by $\mu > 0$; each of these approximate problems have stronger theoretical properties than the problem with $\mu = 0$ (Ralph and Wright, 2004). A solution procedure whereby μ is successively reduced can be implemented as a simple option file to NLPEC, and this has proven remarkably effective. Further details can be found in the NLPEC documentation (GAMS Development Corporation, 2008). The approach has been used to effectively optimize the rig in a sailboat design (Wallace et al., 2006) and to solve a variety of distillation optimization problems (Baumrucker et al., 2008).

It is also possible to generalize the above complementarity condition to a mixed complementarity condition; details can be found in Ferris et al. (2005). Underlying the NLPEC \solver package" is an automatic conversion of the original problem into a standard nonlinear program which is carried out at a scalar model level. The technology to perform this conversion forms the core of the codes that we use to implement the model extensions of the sequel.

2.3. Variational Inequalities

A variational inequality VI(F, X) is to find $x \in X$:

$$F(x)^T(z - x) \geq 0, \text{ for all } z \in X.$$

Here X is a closed (frequently assumed convex) set, defined for example as

$$X = \{x \mid x \geq 0, h(x) \leq 0\}. \tag{9}$$

Note that the first-order (minimum principle) conditions of a nonlinear program

$$\min_{z \in X} f(z)$$

are precisely of this form with $F(x) = \nabla f(x)$. For a concrete example, note that these conditions are necessary and sufficient for the optimality of a linear programming problem: solving the linear program (2) is equivalent to solving the variational inequality given by

$$F(x) = c, \quad X = \{x \mid Ax \geq b, x \geq 0\}. \tag{10}$$

In this case, F is simply a constant function. While there are a large number of instances of the problem that arise from optimization applications, there are many cases where F is not the gradient of any function f. For example, asymmetric traffic equilibrium problems have this format, where the asymmetry arises for example due to different costs associated with left or right hand turns. A complete treatment of the theory and algorithms in this domain can be found in Facchinei and Pang (2003).

Variational inequalities are intimately connected with the concept of a normal cone to a set S, for which a number of authors have provided a rich calculus. Instead of overloading a reader with more notation, however, we simply refer to the seminal work in this area, Rockafellar and Wets (1998). While the theoretical development of this area is very rich, the practical application has been somewhat limited. The notable exception to this is in traffic analysis, see Harker (1993).

It is well known that such problems can be reformulated as complementarity problems when the set X has the representation (9) by introducing multipliers λ on the constraints h:

$$\begin{array}{lcl} 0 \leq F(x) + \lambda^T \nabla h(x) & \perp & x \geq 0 \\ 0 \leq -h(x) & \perp & \lambda \geq 0. \end{array}$$

If X has a different representation, this construction would be modified appropriately. In the linear programming example (10), these conditions are precisely those already given as (3).

When X is the nonnegative orthant, the VI is just an alternative way to state a complementarity problem. However, when X is a more general set, it may be possible to treat it differently than simply introducing multipliers, see Cao and Ferris (1996) for example. In particular, when X is a polyhedral set, algorithms may wish to generate iterates via projection onto X.

A simple two dimensional example may be useful to improve understanding. Let

$$F(x) = \begin{bmatrix} x_1 + 2 \\ x_1 + x_2 - 3 \end{bmatrix}, \; X = \{x \geq 0 \mid x_1 + x_2 \leq 1\},$$

so that F is an affine function, but F is not the gradient of any function $f : \mathbf{R}^2 \to \mathbf{R}$. For this particular data, VI(F, X) has a unique solution

$x = (0, 1)$. Such a variational inequality can be described in GAMS via the model statement

```
model vi / F, defh /;
solve vi using emp;
```

combined with an annotation file that indicates certain equations are to be treated differently by the EMP tool. In this case, the \empinfo" file

```
viFunc
F('1') x('1')
F('2') x('2')
```

identifies that the model equations F define a function F that is to be part of a variational inequality, while the equations defh define constraints of X. Details on specific syntax can be found in GAMS Development Corporation (2009).

2.4. Bilevel Programs

Mathematical programs with optimization problems in their constraints have a long history in operations research including Bracken and McGill (1973); Fortuny-Amat and McCarl (1981); Bard (1998). Hierarchical optimization has recently become important in chemical engineering for a number of different applications as outlined in the introduction. New codes are being developed that exploit this structure, at least for simple hierarchies, and attempt to define and implement algorithms for their solution.

The simplest case is that of bilevel programming, where an upper level problem depends on the solution of a lower level optimization. For example:

$$\begin{aligned} &\min_{x,y} f(x,y) \\ &\text{s.t. } g(x,y) \leq 0, \\ &\qquad y \text{ solves } \min_y v(x,y) \text{ s.t. } \quad h(x,y) \leq 0. \end{aligned}$$

This problem can be reformulated as an MPCC by replacing the lower level optimization problem by its optimality conditions:

$$\begin{aligned} &\min_{x,y} f(x,y) \\ &\text{s.t. } g(x,y) \leq 0, \\ &\qquad 0 = \nabla_y v(x,y) + \lambda^T \nabla_y h(x,y) \perp x \text{ free} \\ &\qquad 0 \leq -h(x,y) \perp \lambda \geq 0. \end{aligned}$$

This approach then allows such problems to be solved using the NLPEC code, for example. However, there are several possible deficiencies that should be noted. Firstly, the optimality conditions encompassed in the complementarity constraints may not have a solution, or the solution may only be necessary (and not sufficient) for optimality. Secondly, the MPCC solver may only find local solutions to the problem. The quest for practical optimality conditions and robust global solvers remains an active area of research. Importantly, the EMP tool will provide the underlying structure of the model to a solver if these advances determine appropriate ways to exploit this.

We can model this bilevel program in GAMS by

```
model bilev /deff,defg,defv,defh/;
solve bilev using emp min f;
```

along with some extra annotations to a subset of the model defining equations. Specifically, within an \empinfo" file we state that the lower level problem involves the objective v which is to be minimized subject to the constraints specified in defv and defh.

```
bilevel x
min v defv defh
```

Note that the variables x are declared to be variables of the upper level problem and that defg will be an upper level constraint. The specific syntax is described in GAMS Development Corporation (2009). Having written the problem in this way, the MPCC is generated automatically, and passed on to a solver. In the case where that solver is NLPEC, a further reformulation of the model is carried out to convert the MPCC into an equivalent NLP or a parametric sequence of NLP's.

A point that has been glossed over here but which is described carefully in the user manual is the process whereby multiple lower level problems are specified. Thus, the EMP model type allows multiple lower level problems to be specified within the bilevel format.

2.5. Mathematical Programs with Equilibrium Constraints

Mathematical programs with equilibrium constraints are a generalization of the aforementioned MPCC problem class. The difference is that the lower level problem, instead of being a complementarity problem, is now a variational inequality. To specify such a model we couple together the approaches of the last two subsections. Thus, the MPEC

$$\begin{aligned} \min_{x,y} \ & f(x,y) \\ \text{s.t.} \ & g(x,y) \leq 0, \\ & y \text{ solves VI}(F(x,\cdot), X), \end{aligned}$$

where $X = \{s \mid h(x,s) \leq 0\}$ can be specified by the statements:

```
model mpecmod /deff,defg,F,defh/;
solve mpecmod using emp min f;
```

and an \empinfo" file

```
bilevel x
vi F defh
vifunc
  F('1') y('1')
  F('2') y('2')
  ...
```

Thus, the EMP format allows for Stackelberg games to be formulated where there are collections of second level players, and each of those second level players could be solving a variational inequality or an optimization problem. The EMP model type results in the ability to model and solve such problems.

3. Extended Nonlinear Programs

Optimization models have traditionally been of the form (1). Specialized codes have allowed certain problem structures to be exploited algorithmically, for example simple bounds on variables. However, for the most part, assumptions of smoothness of f, g and h are required for many solvers to process these problems effectively.

In a series of papers, Rockafellar and colleagues (Rockafellar, 1993, 1999, 1987) have introduced the notion of extended nonlinear programming, where the (primal) problem has the form:

$$\min_{x \in X} f_0(x) + \theta(f_1(x), \ldots, f_m(x)). \tag{11}$$

In this setting, X is assumed to be a nonempty polyhedral set, and the functions $f_0, f_1, \ldots, f_m$ are smooth. The function θ can be thought of as a generalized penalty function that may well be nonsmooth. However, when θ has the following form

$$\theta(u) = \sup_{y \in Y} \{y^T u - k(y)\}, \tag{12}$$

a computationally exploitable and theoretically powerful framework can be developed based on conjugate duality. A key point for computation and modeling is that the function θ can be fully described by defining the set Y and the function k. Furthermore, as we show below, different choices lead to a rich variety of functions θ, many of which are extremely useful for modeling. In the above setting θ can take on the value of ∞ and may well be nonsmooth, but it is guaranteed to be convex (proper and lower semicontinuous when $Y \neq \emptyset$ and k is smooth and convex).

Furthermore, from a modeling perspective, an extended nonlinear program can be specified simply by defining the functions $f_0, f_1, \ldots, f_m$ in the manner already provided by the modeling system, with the additional issue of simply defining Y and k. Conceptually, this is not much harder that what is carried out already, but leads to significant enhancements to the types of models that are available. This paper outlines an approach to do this within the GAMS modeling system for a number of different choices of Y and k.

The EMP model type works in this setting by providing a library of functions θ that specify a variety of choices for k and Y. Once a modeler determines which constraints are treated via which choice of k and Y, the EMP model interface automatically forms an equivalent variational inequality or complementarity problem. As we show later, there may be alternative formulations that are computationally more appealing; such reformulations can be generated using different options to our tool.

3.1. Forms of θ

The EMP model type makes the problem format (11) available to users in GAMS. As special cases, we can model piecewise linear penalties, least squares and L_1 approximation problems, as well as the notion of soft and hard constraints. We allow modelers to utilize cone constraints and pass on the underlying geometric structure to solvers. Particular examples show enormous promise both from a modeling and solution perspective.

For ease of exposition, we now describe a subset of the types of functions θ that can be generated by particular choices of Y and k. In many cases, the function θ is separable, that is

$$\theta(u) = \sum_{i=1}^{m} \theta_i(u_i).$$

so we can either specify θ_i or θ itself.

Extended nonlinear programs include the classical nonlinear programming form (1) as a special case. This follows from the observation that if K

is a closed convex cone, and we let ψ_K denote the \indicator function" of K defined by:

$$\psi_K(u) = \begin{cases} 0 & \text{if } u \in K \\ \infty & \text{else,} \end{cases}$$

then (1) can be rewritten as:

$$\min_x f(x) + \psi_K((g(x), h(x)), \; K = \mathbf{R}^m_- \times \{0\}^p,$$

where m and p are the dimensions of g and h respectively and $\mathbf{R}^m_- = \{u \in \mathbf{R}^m \mid u \leq 0\}$. An elementary calculation shows that

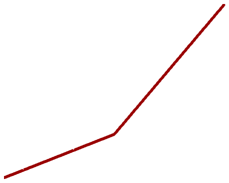
$$\psi_K(u) = \sup_{v \in K^\circ} u^T v,$$

where $K^\circ = \left\{u \mid u^T v \leq 0, \forall v \in K\right\}$ is the polar cone of the given cone K. Thus, when $\theta(u) = \psi_K(u)$ we simply take

$$k \equiv 0 \text{ and } Y = K^\circ. \tag{13}$$

In our example, $K^\circ = \mathbf{R}^m_+ \times \mathbf{R}^p$. To some extent, this is just a formalism that allows us to claim the classical case as a specialization; however when we take the cone K to be more general than the polyhedral cone used above, we can generate conic programs (see below) for example.

The second example involves a piecewise linear function θ:



Formally, for $u \in \mathbf{R}$,

$$\theta(u) = \begin{cases} \rho u & \text{if } u \geq 0 \\ \sigma u & \text{else.} \end{cases}$$

In this case, simple calculations prove that θ has the form (12) for the choices:

$$k \equiv 0 \text{ and } Y = [\sigma, \rho].$$

The special case where $\sigma = -\rho$ results in

$$\theta(u) = \rho\,|u|\,. \tag{14}$$

This allows us to model nonsmooth L_1 approximation problems. Another special case results from the choice of $\sigma = 0$, whereby

$$\theta(u) = \rho \max\{u, 0\}.$$

This formulation corresponds to a soft penalization on an inequality constraint, namely if $\theta(f_1(x))$ is used then nothing is added to the objective function if $f_1(x) \leq 0$, but $\rho f_1(x)$ is added if the constraint $f_1(x) \leq 0$ is violated. Contrast this to the classical setting above, where ∞ is added to the objective if the inequality constraint is violated. It is interesting to see that truncating the set Y, which amounts to bounding the multipliers, results in replacing the classical constraint by a linearized penalty.

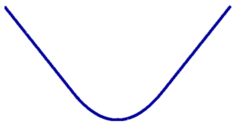
The third example involves a more interesting choice of k. If we wish to replace the \absolute value" penalization given above by a quadratic penalization (as in classical least squares analysis), that is

$$\theta(u) = \gamma u^2 \tag{15}$$

then a simple calculation shows that we should take

$$k(y) = \frac{1}{4\gamma}y^2 \text{ and } Y = \mathbf{R}.$$

By simply specifying this different choice of k and Y we can generate such models easily and quickly within the modeling system: note however that the reformulation we would use in (14) and (15) are very different as we shall explain in the simple example below. Furthermore, in many applications it has become popular to penalize violations using a quadratic penalty only within a certain interval, afterwards switching to a linear penalty (chosen to make the penalty function θ continuously differentiable - see Huber (1981). That is:



$$\text{i.e. } \theta(u) = \begin{cases} \gamma u - \frac{1}{2}\gamma^2 & \text{if } u \geq \gamma \\ \frac{1}{2}u^2 & \text{if } u \in [-\gamma, \gamma] \\ -\gamma u - \frac{1}{2}\gamma^2 & \text{else.} \end{cases}$$

Such functions arise from quadratic k and simple bound sets Y. In particular, the somewhat more general function

$$\theta(u) = \begin{cases} \gamma\beta^2 + \rho(u - \beta) & \text{if } u \geq \beta \\ \gamma u^2 & \text{if } u \in [\alpha, \beta] \\ \gamma\alpha^2 + \sigma(u - \alpha) & \text{else} \end{cases}$$

arises from the choice of

$$k(y) = \frac{1}{4\gamma}y^2 \text{ and } Y = [\sigma, \rho],$$

with $\alpha = \frac{\sigma}{2\gamma}$ and $\beta = \frac{\rho}{2\gamma}$.

The final example that we give is that of L_∞ penalization. This example is different to the examples given above in that θ is not separable. However, straightforward calculation can be used to show

$$\theta(u) = \max_{i=1,\ldots,m} u_i$$

results from the choice of

$$k \equiv 0 \text{ and } Y = \left\{ y \in \mathbf{R}^m \mid y \geq 0, \sum_{i=1}^{m} y_i = 1 \right\},$$

that is, Y is the unit simplex.

3.2. Underlying theory

The underlying structure of θ leads to a set of extended optimality conditions and an elegant duality theory. This is based on an extended form of the Lagrangian:

$$\mathcal{L}(x, y) = f_0(x) + \sum_{i=1}^{m} y_i f_i(x) - k(y)$$
$$x \in X, y \in Y$$

Note that the Lagrangian $\mathcal{L}$ is smooth - all the nonsmoothness is captured in the θ function. The theory is an elegant combination of calculus arguments related to f_i and its derivatives, and variational analysis for features related to θ.

It is shown in Rockafellar (1993) that under a standard constraint qualification, the first-order conditions of (11) are precisely in the form of the following variational inequality:

$$\text{VI}\left(\begin{bmatrix}\nabla_x\mathcal{L}(x,y)\\ -\nabla_y\mathcal{L}(x,y)\end{bmatrix}, X\times Y\right). \tag{16}$$

When X and Y are simple bound sets, this is simply a complementarity problem.

Note that EMP exploits this result. In particular, if an extended nonlinear program of the form (11) is given to EMP, then the optimality conditions (16) are formed as a variational inequality problem and can be processed as outlined above. For a specific example, we cite the fact that if we use the (classical) choice of k and Y given in (13), then the optimality conditions of (11) are precisely the standard complementarity problem given as (5). While this is of interest, we believe that other choices of k and Y may be more useful and lead to models that have more practical significance.

Under appropriate convexity assumptions on this Lagrangian, it can be shown that a solution of the VI (16) is a saddle point for the Lagrangian on $X \times Y$. Furthermore, in this setting, the saddle point generates solutions to the primal problem (11) and its dual problem:

$$\max_{y\in Y} g(y), \quad \text{where } g(y) = \inf_{x\in X}\mathcal{L}(x,y),$$

with no duality gap.

3.3. A simple example

As an example, consider the problem

$$\begin{aligned} \min_{x_1,x_2,x_3} \quad & \exp(x_1) + 5\left\|\log(x_1) - 1\right\|^2 + 2\max(x_2^2 - 2, 0) \\ \text{s.t.} \quad & x_1/x_2 = \log(x_3), \\ & 3x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

In this problem, we would take

$$X = \left\{x \in \mathbf{R}^3 \mid 3x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0\right\}.$$

The function θ essentially treats 3 separable pieces:

$$\begin{aligned} f_1(x) &= \log(x_1) - 1, \\ f_2(x) &= x_2^2 - 2, \\ f_3(x) &= x_1/x_2 - \log(x_3). \end{aligned}$$

A classical problem would force $f_1(x) = 0$, $f_2(x) \leq 0$ and $f_3(x) = 0$, while minimizing $f_0(x) = \exp(x_1)$. In our problem, we still force $f_3(x) = 0$, but apply a (soft) least squares penalty on $f_1(x)$ and a smaller one-sided penalization on $f_2(x)$. The above formulation is nonsmooth due to the max term in the objective function; in practice we could replace this by:

$$\begin{aligned} \min_{x_1,x_2,x_3,w} \quad & \exp(x_1) + 5\left\|\log(x_1) - 1\right\|^2 + 2w \\ \text{s.t.} \quad & x_1/x_2 = \log(x_3), \\ & 3x_1 + x_2 \leq 5, x_1 \geq 0, x_2 \geq 0 \\ & w \geq x_2^2 - 2, w \geq 0 \end{aligned}$$

and recover a standard form NLP. If the penalty on $f_1(x)$ would be replaced by a one-norm penalization (instead of least squares), we would have to play a similar game, moving the function $f_1(x)$ into the constraints and adding additional variable(s). To some extent, this seems unnatural - a modeler should be able to interchange the penalization without having to reformulate the problem from scratch.

The proposed extended NLP would not be reformulated at all by the modeler, but allows all these \generalized constraints" to be treated in a similar manner within the modeling system. The actual formulation would take:

$$\theta(u) = \theta_1(u_1) + \theta_2(u_2) + \theta_3(u_3)$$

where

$$\begin{aligned} \theta_1(u_1) &= 5u_1^2, \\ \theta_2(u_2) &= 2\max(u_2, 0), \\ \theta_3(u_3) &= \psi_{\{0\}}(u_3). \end{aligned}$$

The discussion above allows us to see that

$$\begin{aligned} Y &= \mathbf{R} \times [0, 2] \times \mathbf{R}, \\ k(y) &= \frac{1}{20}y_1^2 + 0 + 0. \end{aligned}$$

The corresponding Lagrangian is the smooth function:

$$\mathcal{L}(x, y) = f_0(x) + \sum_{i=1}^{3} y_i f_i(x) - k(y).$$

The corresponding VI (16) can almost be formulated in GAMS (except that the linear constraint in X cannot be handled currently except by introducing a $\theta_4(x)$). Thus

$$f_4(x) = 3x_1 + x_2 - 5,\ \theta_4(u) = \psi_{\mathbf{R}_-}$$

resulting in the following choices for Y and k:

$$\begin{aligned} Y &= \mathbf{R} \times [0,2] \times \mathbf{R} \times \mathbf{R}_+, \\ k(y) &= \frac{1}{20}y_1^2 + 0 + 0 + 0. \end{aligned}$$

Since X and Y are now simple bound sets, (16) is now a complementarity problem and can be solved for example using PATH. A simple \empinfo" file details the choices of Y and k from the implemented library:

```
Adjusteqn
e1 sqr 5
e2 MaxZ 2
```

The full model and option files are available in GAMS Development Corporation (2009).

3.4. Reformulation as a classical NLP

Suppose

$$\theta(u) = \sup_{y \in Y}\{u^T y - \frac{1}{2}y^T Q y, \}$$

for a polyhedral set $Y \in \mathbf{R}^m$ and a symmetric positive semidefinite $Q \in \mathbf{R}^{m \times m}$ (possibly $Q = 0$). Suppose further that

$$\begin{aligned} X &= \{x \mid Rx \leq r\}, \ \ Y = \left\{y \mid S^T y \leq s\right\}, \\ Q &= DJ^{-1}D^T, \ \ F(x) = (f_1(x), \ldots, f_m(x)), \end{aligned}$$

where J is symmetric and positive definite (for instance $J = I$). Then, as outlined by Rockafellar (1999), the optimal solutions x of (11) are the x components of the optimal solutions (x, z, w) to

$$\begin{array}{ll} \text{min} & f_0(x) + s^T z + \frac{1}{2} w^T J w \\ \text{s.t.} & Rx \leq r, z \geq 0, F(x) - Sz - Dw = 0. \end{array}$$

The multiplier on the equality constraint in the usual sense is the multiplier associated with x in the extended Lagrangian for (11). (Note that a Cholesky factorization may be needed to determine D.)

It may be better to solve this reformulated NLP than to solve (16). However, it is important that we can convey all types of nonsmooth optimization problems to a solver as smooth optimization problems, and hence it is important to communicate the appropriate structure to the solver interface. We believe that specifying Y and k is a theoretically sound way to do this.

Another example showing formulation of an extended nonlinear program as a complementarity problem within GAMS can be found in Dirkse and Ferris (1995).

4. More Allowable Model Constructs

4.1. Disjunctive Programs

There are many ways that the EMP model type can be used for further extensions to the modeling capabilities of a given system. In particular, the procedures outlined in Vecchietti et al. (2003) for disjunctive programming extensions are also implemented within the EMP model type.

One simple example to highlight this feature is the notion of an ordering of tasks, namely that either job i comes before job j or the converse. Such a disjunction can be specified using an empinfo file containing lines:

```
  disjuncton * seq(i,j) else seq(j,i)
```

In such an example, one can implement a Big-M method, employ indicator constraints, or utilize a convex hull reformulation. The convex hull reformulation is the default strategy; to utilize the Big-M formulation, the additional option

```
  default bigm 1000
```

would add binary variables and constraints to impose the disjunction using a Big-M value of 1000. Alternatively, for the CPLEX solver, the option setting (for EMP):

```
  default indic
```

writes out a model and a CPLEX option file that implements a reformulation using indicator constraints. The EMP model library that is part of the standard GAMS distribution contains a sequencing model that implements all of these options.

More complicated (nonlinear) examples make the utility of this approach clearer. The design of a multiproduct batch plan with intermediate storage described in Vecchietti and Grossmann (1999) and a synthesis problem involving 8 processes from Turkay and Grossmann (1996) are also included in the EMP model library. As a final example, the gasoline emission model outlined in Furman and Androulakis (2008) is precisely in the form that could exploit the features of EMP related to (nonlinear) disjunctive programming.

4.2. Conic Programming

A problem of significant recent interest (due to its applications in robust optimization and optimal control) involves conic constraints (Lobo et al., 1998; Alizadeh and Goldfarb, 2003; Ben-Tal and Nemirovskii, 2001):

$$\min_{x \in X} p^T x \text{ s.t. } Ax - b \leq 0, x \in C,$$

where C is a convex cone. Using the notation outlined above, this can be expressed as an EMP:

$$\min_{x \in X} p^T x + \psi_{\mathbf{R}^m_-}(Ax - b) + \psi_C(x)$$

For specific cones such as the Lorentz (ice-cream) cone where $C = \left\{ x \in \mathbf{R}^n \mid x_1 \geq \sqrt{\sum_{i=2}^n x_i^2} \right\}$, or the rotated quadratic cone, there are efficient implementations of interior point algorithms for their solution (Andersen et al., 2003). It is also possible to reformulate the problem in the form (1) for example by adding the constraint

$$x_1 \geq \sqrt{\sum_{i=2}^n x_i^2}. \tag{17}$$

Annotating the variables that must lie in a particular cone using a \empinfo" file allows solvers like MOSEK (Andersen and Andersen, 2000) to receive the problem as a cone program, while standard NLP solvers would see a reformulation of the problem as a nonlinear program. It is also easy to see that (17) can be replaced by the following equivalent constraints

$$x_1^2 \geq \sum_{i=2}^n x_i^2, \; x_1 \geq 0.$$

Such constraints can be added to a nonlinear programming formulation or a quadratically constrained (QCP) formulation in GAMS. This automatic reformulation allows the CPLEX solver to process these problems since its barrier solver will process constraints of the form

$$y^2 \geq x^T Qx, \; y \geq 0, Q \text{ PSD.}$$

Details on the options that implement these approaches can be found in the EMP manual.

Furthermore, it is straightforward to facilitate the use of stochastic constraints that have become very popular in financial applications. Specifically, we mention the work of Rockafellar and Uryasev (2000) on conditional value at risk, and the recent papers by Dentcheva and Ruszczynski (2003), and Luedtke (2008) on stochastic dominance constraints. All of these formulations are easily cast as constraints on decision variables annotated by additional (in this case distributional) information.

4.3. Embedded Complementarity Systems

A different type of embedded optimization model that arises frequently in applications is:

$$\begin{array}{ll} \min\limits_{x} & f(x,y) \\ \text{s.t.} & g(x,y) \leq 0 \quad (\perp \lambda \geq 0) \end{array}$$

$$H(x,y,\lambda) = 0 \qquad (\perp y \text{ free})$$

Note the difference here: the optimization problem is over the variable x, and is parameterized by the variable y. The choice of y is fixed by the (auxiliary) complementarity relationships depicted here by H. Note that the $\backslash H$" equations are not part of the optimization problem, but are essentially auxiliary constraints to tie down remaining variables in the model.

Within GAMS, this is modeled as:

```
model ecp /deff,defg,defH/;
solve ecp using emp;
```

Again, so this model can be processed correctly as an EMP, the modeler provides additional annotations to the model defining equations in an \empinfo" file, namely that the function H that is defined in defH is complementary to the variable y (and hence the variable y is a parameter to the optimization problem), and furthermore that the dual variable associated with the equation defg in the optimization problem is one and the same as the variable λ used to define H:

```
dualequ H y
dualvar lambda g
```

Armed with this additional information, the EMP tool automatically creates the following MCP:

$$\begin{array}{lll} 0 = \nabla_x \mathcal{L}(x,y,\lambda) & \perp & x \text{ free} \\ 0 \leq -\nabla_\lambda \mathcal{L}(x,y,\lambda) & \perp & \lambda \geq 0 \\ 0 = H(x,y,\lambda) & \perp & y \text{ free}, \end{array}$$

where the Lagrangian is defined as

$$\mathcal{L}(x,y,\lambda) = f(x,y) + \lambda^T g(x,y).$$

Perhaps the most popular use of this formulation is where competition is allowed between agents. A standard method to deal with such cases is via the concept of Nash Games. In this setting x^* is a Nash Equilibrium if

$$x_i^* \in \arg\min_{x_i \in X_i} \ell_i(x_i, x_{-i}^*, q), \forall i \in \mathcal{I},$$

where x_{-i} are other players decisions and the quantities q are given exogenously, or via complementarity:

$$0 \leq H(x, q) \quad \perp \quad q \geq 0.$$

This mechanism is extremely popular in economics, and Nash famously won the Nobel Prize for his contributions to this literature.

This format is again an EMP, more general than the example given above in two respects. Firstly, there is more than one optimization problem specified in the embedded complementarity system. Secondly, the parameters in each optimization problem consist of two types. Firstly, there are the variables q that are tied down by the auxiliary complementarity condition and hence are treated as parameters by the ith Nash player. Also there are the variables x_{-i} that are treated as parameters by the ith Nash player, but are treated as variables by a different player j. While we do not specify the syntax here for these issues, GAMS Development Corporation (2009) provides examples that outline how to carry out this matching within GAMS. Finally, two points of note: first it is clear that the resulting model is a complementarity problem and can be solved using PATH, for example. Secondly, performing the conversion from an embedded complementarity system or a Nash Game automatically is a critical step in making such models practically useful.

We note that there is a large literature on discrete-time finite-state stochastic games: this has become a central tool in analysis of strategic interactions among forward-looking players in dynamic environments. The Ericson and Pakes (1995) model of dynamic competition in an oligopolistic industry is exactly in the format described above, and has been used extensively in applications such as advertising, collusion, mergers, technology adoption, international trade and finance. Ongoing work aims to use the EMP format to model these problems.

5. Conclusions

A number of new modeling formats involving complementarity and variational inequalities have been described and a framework, EMP, that allows such problems to be specified has been outlined. We believe this will make a modeler's task easier by allowing model structure to be described succinctly. Furthermore, model generation can be done more reliably and automatically, and algorithms can exploit model structure to improve solution speed and robustness.

We believe that EMP will be useful in systems optimization where collections of interacting (optimization and complementarity) models need to be processed. The automatic reformulations of these problems will save time, improve accuracy, and expand the range of problems that can be practicably solved. While the implementation described in this paper was developed for the GAMS modeling system, we believe that most of the features are applicable to any modeling system. Specifically, equation annotations can be provided in AMPL via user-defined constructs while the object oriented design of MATLAB could easily be extended to this setting.

References

Alizadeh, F. and Goldfarb, D. (2003). Second-order cone programming. *Mathematical Programming*, 95:3{51.

Andersen, E. D. and Andersen, K. D. (2000). The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In et al., H. F., editor, *High Performance Optimization*, pages 197{232. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Andersen, E. D., Roos, C., and Terlaky, T. (2003). On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249{277.

Bard, J. F. (1998). *Practical Bilevel Optimization: Algorithms and Applications*, volume 30 of *Nonconvex Optimization and its Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Baumrucker, B. T., Renfro, J. G., and Biegler, L. T. (2008). MPEC problem formulations and solution strategies with chemical engineering applications. *Computers and Chemical Engineering*, 32:2903{2913.

Ben-Tal, A. and Nemirovskii, A. (2001). *Lectures on Modern Convex Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, PA.

Bisschop, J. and Entriken, R. (1993). *AIMMS – The Modeling System*. Paragon Decision Technology, Haarlem, The Netherlands.

Bisschop, J. and Meeraus, A. (1982). On the development of a general algebraic modeling system in a strategic planning environment. *Mathematical Programming Study*, 20:1{29.

Bracken, J. and McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research*, 21:37{44.

Brooke, A., Kendrick, D., and Meeraus, A. (1988). *GAMS: A User's Guide*. The Scientific Press, South San Francisco, California.

Burgard, A. P. and Maranas, C. D. (2003). Optimization-based framework for inferring and testing hypothesized metabolic objective functions. *Biotechnology and Bioengineering*, 82(6):670{677.

Burgard, A. P., Pharkya, P., and Maranas, C. D. (2003). Optknock: A bilevel programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and Bioengineering*, 84(6):647{657.

Cao, M. and Ferris, M. C. (1996). A pivotal method for affine variational inequalities. *Mathematics of Operations Research*, 21:44{64.

Clark, P. A. and Westerberg, A. W. (1990). Bilevel programming for steady state chemical process design: Fundamentals and algorithms. *Computers and Chemical Engineering*, 14:87{97.

Dash Optimization (2009). XPress-mosel. http://www.dashoptimization.com/products/products_mosel.html.

Dentcheva, D. and Ruszczynski, A. (2003). Optimization with stochastic dominance constraints. *SIAM Journal on Optimization*, 14:548{566.

Dirkse, S. P. and Ferris, M. C. (1995). MCPLIB: A collection of nonlinear mixed complementarity problems. *Optimization Methods and Software*, 5:319{345.

Ericson, R. and Pakes, A. (1995). Markov perfect industry dynamics: A framework for empirical analysis. *Review of Economic Studies*, 62:53{82.

Facchinei, F. and Pang, J. S. (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer-Verlag, New York, New York.

Ferris, M. C., Dirkse, S. P., and Meeraus, A. (2005). Mathematical programs with equilibrium constraints: Automatic reformulation and solution via constrained optimization. In Kehoe, T. J., Srinivasan, T. N., and Whalley, J., editors, *Frontiers in Applied General Equilibrium Modeling*, pages 67{93. Cambridge University Press.

Ferris, M. C., Fourer, R., and Gay, D. M. (1999). Expressing complementarity problems and communicating them to solvers. *SIAM Journal on Optimization*, 9:991{1009.

Ferris, M. C. and Munson, T. S. (2000). Complementarity problems in GAMS and the PATH solver. *Journal of Economic Dynamics and Control*, 24:165{188.

Ferris, M. C. and Pang, J. S. (1997). Engineering and economic applications of complementarity problems. *SIAM Review*, 39:669{713.

Fortuny-Amat, J. and McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 32(9):783{792.

Fourer, R., Gay, D. M., and Kernighan, B. W. (1990). A modeling language for mathematical programming. *Management Science*, 36:519{554.

Fourer, R., Gay, D. M., and Kernighan, B. W. (1993). *AMPL: A Modeling*

Language for Mathematical Programming. Duxbury Press, Pacific Grove, California.

Furman, K. and Androulakis, I. P. (2008). A novel MINLP-based representation of the original complex model for predicting gasoline emissions. *Computers and Chemical Engineering*, 32:2857{2876.

GAMS Development Corporation (2008). *NLPEC,* User's Manual. http://www.gams.com/solvers/nlpec.pdf.

GAMS Development Corporation (2009). *EMP,* User's Manual. http://www.gams.com/solvers/emp.pdf.

Gopal, V. and Biegler, L. T. (1997). Nonsmooth dynamic symulation with linear programming based methods. *Computers and Chemical Engineering*, 21(7):675{689.

Gopal, V. and Biegler, L. T. (1999). Smoothing methods for complementarity problems in process engineering. *Journal of AIChE*, 45(7):1535{1547.

Harker, P. T. (1993). *Lectures on Computation of Equilibria with Equation–Based Methods*. CORE Lecture Series. CORE Foundation, Louvain{la{Neuve, Universite Catholique de Louvain.

Huber, P. J. (1981). *Robust Statistics*. John Wiley & Sons, New York.

Lobo, M., Vandenberghe, L., Boyd, S., and Lebret, H. (1998). Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193{228.

Luedtke, J. (2008). New formulations for optimization under stochastic dominance comstraints. *SIAM Journal on Optimization*, 19:1433{1450.

Maximal Software (2009). MPL. http://www.maximal-usa.com/mpl/what.html.

Mitsos, A., Lemonidis, P., and Barton, P. I. (2008). Global solution of bilevel programs with a nonconvex inner program. *Journal of Global Optimization*, 42:475{513.

Raghunathan, A. U., Perez-Correa, J. R., Agosin, E., and Biegler, L. T. (2006). Parameter estimation in metabolic flux balance models for batch fermentation-formulation and solution using differential variational inequalities. *Annals of Operations Research*, 148:251{270.

Ralph, D. and Wright, S. J. (2004). Some properties of regularization and penalization schemes for MPECs. *Optimization Methods and Software*, 19(5):527{556.

Rockafellar, R. T. (1987). Linear-quadratic programming and optimal control. *SIAM Journal on Control and Optimization*, 25:781{814.

Rockafellar, R. T. (1993). Lagrange multipliers and optimality. *SIAM Review*, 35:183{238.

Rockafellar, R. T. (1999). Extended nonlinear programming. In Pillo, G. D. and Giannnessi, F., editors, *Nonlinear Optimization and Related Topics*, pages 381{399. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value-at-risk. *The Journal of Risk*, 2:21{41.

Rockafellar, R. T. and Wets, R. J. B. (1998). *Variational Analysis*. Springer-Verlag.

Rutherford, T. F. (1999). Applied general equilibrium modeling with MPSGE as a GAMS subsystem: An overview of the modeling framework and syntax. *Computational Economics*, 14:1{46.

Tawarmalani, M. and Sahinidis, N. V. (2004). Global Optimization of Mixed Integer Nonlinear Programs: A Theoretical and Computational Study. *Mathematical Programming*, 99(3):563{591.

Turkay, M. and Grossmann, I. E. (1996). Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering*, 20:959{978.

Van Hentenryck, P. (1999). *The OPL Optimization Programming Language*. MIT Press.

Vecchietti, A. and Grossmann, I. E. (1999). LOGMIP: a disjunctive 0-1 nonlinear optimizaer for process system models. *Computers and Chemical Engineering*, 23:555{565.

Vecchietti, A., Lee, S., and Grossmann, I. E. (2003). Modeling of discrete/continuous optimization problems: Characterizaton and formulations of disjunctions and their relaxations. *Computers and Chemical Engineering*, 27:433{448.

Viswanathan, J. and Grossmann, I. E. (1990). A combined penalty fucntion and outer-approximation method for MINLP optimization. *Computers and Chemical Engineering*, 14:769{782.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25{57.

Wallace, J., Philpott, A., O'Sullivan, M., and Ferris, M. (2006). Optimal rig design using mathematical programming. In *2nd High Performance Yacht Design Conference, Auckland, 14–16 February, 2006*, pages 185{192.

Westerlund, T., Skrifvars, H., Harjunkoski, I., and Pörn, R. (1998). An extended cutting plane method for solving a class of non-convex minlp problems. *Computers and Chemical Engineering*, 22(3):357{365.

Yang, L., Mahadevan, R., and Cluett, W. R. (2008). A bilevel optimization algorithm to identify enzymatic capacity constraints in metabolic networks. *Computers and Chemical Engineering*, 32:2072{2085.