CHALLENGES AND RESEARCH ISSUES FOR PRODUCT AND PROCESS DESIGN OPTIMIZATION

Lorenz T. Biegler and Ignacio E. Grossmann
Chemical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Optimization as an enabling technology has been one of the big success stories in process systems engineering. In this paper we present first a general review of optimization and its applications to a variety of problems in process systems engineering. Next, we provide an overview of two key areas: nonlinear programming and logic-based discrete/continuous optimization. In particular, recent advances are presented in the modeling and solution of nonlinear programming, dynamic optimization, mixed-integer and generalized disjunctive programming, global optimization and constraint programming. The impact of these techniques is illustrated with several example problems.

Keywords Process Optimization, Nonlinear Programming, Mixed Integer Nonlinear Programming, Logic-based Optimization, Dynamic Optimization, Complementarity, Disjunctive Programming, Global Optimization, Constraint Programming.

Introduction

Our objective in this paper is to provide an overview of new developments in nonlinear and discrete/continuous optimization. The emphasis is on large-scale nonlinear programming and logic-based optimization, both of which are relevant for product and process design. Optimization has become a major enabling area in process systems engineering. It that has evolved from a methodology of academic interest into a technology that continues to make significant impact in industry. Before we discuss the applications of optimization, it is useful to present a classification of problem types. It should be noted that this classification is independent of the solution methods. Optimization problems can be classified in terms of continuous and of discrete variables.

The major problems for continuous optimization include linear (LP) and nonlinear programming (NLP). An important subclass of LP is the linear complementarity problem (LCP), while for the NLP it includes quadratic programming (QP) and semidefinite programming (SP). For the latter, an important distinction is also whether the NLP problem is convex or nonconvex, since the latter may give rise to multiple local optima. Another important distinction is whether the problem is assumed to be differentiable or not.

On the other hand, discrete problems are classified into mixed-integer linear programming (MILP) and mixed-integer nonlinear programming (MINLP). For the former an important particular case is when all the variables are integer, which gives rise to an integer programming (IP) problem. This problem in turn can be classified into many special problems (e.g. assignment, traveling salesman, etc.). The MINLP problem also gives rise to special problems, although here the main distinction like in the NLP problem is whether its relaxation is convex or non-convex.

Regarding their formulation, discrete/continuous optimization problems when represented in algebraic form, correspond to mixed-integer optimization problems that have the following general form:

$$\begin{aligned} \min\, & Z = f(x,y) \\ s.t.\quad & h(x,y) = 0 \\ & g(x,y) \le 0 \\ & x \in X,\ y \in \{0,1\}^m \end{aligned} \qquad \text{(MIP)}$$

where $f(x, y)$ is the objective function (e.g. cost), $h(x, y) = 0$ are the equations that describe the performance of the system (material balances, production rates), and $g(x,y) \le 0$ are inequalities that define the specifications or constraints for feasible plans and schedules. The variables x are continuous and generally correspond to state variables, while y are the discrete variables, which

generally are restricted to take 0-1 values to define for instance the assignments of equipment and sequencing of tasks. Problem (MIP) corresponds to a mixed-integer nonlinear program (MINLP) when any of the functions involved are nonlinear. If all functions are linear it corresponds to a mixed-integer linear program (MILP). If there are no 0-1 variables, the problem (MIP) reduces to a nonlinear program (NLP) or linear program (LP) depending on whether or not the functions are linear.

It should be noted that (MIP) problems, and their special cases, may be regarded as steady-state models. Hence, one important extension is the case of dynamic models, which in the case of discrete time models gives rise to multiperiod optimization problems, while for the case of continuous time it gives rise to optimal control problems that contain differential-algebraic equation (DAE) models. Another important extension includes problems under uncertainty, which give rise to stochastic optimization problems.

Applications in Process Engineering

Mathematical programming, and optimization in general, have found extensive use in process systems engineering. A major reason for this is that in these problems there are often many alternative solutions, and hence, it is often not easy to find the optimal solution. Furthermore, in many cases the economics is such that finding the optimum solution translates into large savings. Therefore, there might be a large economic penalty to just sticking to suboptimal solutions. In summary, optimization has become a major technology that helps companies to remain competitive.

Applications in Process Design and Synthesis have been dominated by NLP and MINLP models due to the need for the explicit handling of performance equations, although simpler targeting models in process synthesis can give rise to LP and MILP problems. Operations problems, in contrast, tend to be dominated by linear models, LP and MILP, for planning, scheduling and supply chain problems. NLP, however, plays a crucial role at the level of real time optimization. Control has traditionally relied on LP and NLP models, although MILPs are being increasingly used for hybrid systems. Finally, note that global optimization has concentrated more on design than on operations problems, since nonconvexities in the design problems are more likely to yield suboptimal solutions since the corresponding bounds for the variables are rather loose in these problems. It is also worth noting that all of these applications have been facilitated not only by progress in optimization algorithms, but also by the advent of modeling techniques (Williams, 1985) and systems such as GAMS (Brooke et. al, 1998), AMPL (Fourer et al., 1992) and AIMMS (Bisschop and Entriken, 1993).

Here we concentrate on two broad areas that are very active research topics and will strongly influence the future of optimization algorithms and formulations. First, we discuss recent developments of efficient NLP methods particularly for large-scale problems and ill-posed characteristics, such as singular Hessians, dependent constraints and complementarity constraints. We describe recent progress in the development of efficient algorithms that address these issues. Moreover, for large-scale NLPs we describe a number of relevant applications including the treatment of DAE and PDE models. Several large-scale examples are described to illustrate these developments.

Next, we describe new developments in discrete/continuous logic-based optimization. We provide an overview of Generalized Disjunctive Programming (GDP) and its relation with MINLP. We describe several algorithms for GDP that include branch and bound, decomposition and mixed-integer reformulations. We also describe recent developments for cutting plane techniques, global optimization of nonconvex GDP problems, and constraint programming. Several examples are presented to illustrate the capabilities of these methods.

The optimization strategies described in the following sections will not overly emphasize global optimization strategies, as these are already covered by Pardalos and Schoen (2004) in this conference. Moreover, several other papers in this conference discuss applications of optimization in process engineering. Instead, this paper will emphasize NLP and MINLP optimization methods, strategies and concepts as a core area for research in process systems engineering. As a result, this review also serves as a complement to detailed optimization models in specific applications areas that are presented in other papers in this conference.

Advances in Nonlinear Programming Methods

Nonlinear programming algorithms play an important role in numerous process and product applications. They are widely used in the design of chemical processes and the development of new products. This also includes related problems of state and parameter estimation for model building and model discrimination. On the operations side, nonlinear programming is the key component in real time optimization and also in the related problem of data reconciliation and gross error detection. Finally, there are a number of nonlinear control applications that rely on efficient nonlinear programming solvers, both for state estimation as well as the solution of moving horizon problems for model predictive control.

In addition to NLP applications, NLP subproblems arise frequently in dealing with MINLPs and in global

solution strategies. NLP solvers are therefore an important component of algorithms for MINLP (e.g., Outer Approximation, Generalized Benders Decomposition and nonlinear branch and bound algorithms) and global optimization (e.g., spatial branch and bound, DCF, αBB). Nevertheless, for most NLP applications there is a heavy reliance on 'off-the-shelf' solvers (e.g., NPSOL, MINOS, CONOPT, SNOPT) that are bundled within modeling environments like GAMS, AIMMS and AMPL. While these are well written codes, they have a 'one size fits all' approach which becomes inefficient and even unsuccessful for challenging large-scale problems in process engineering. To counter these difficulties, we explore some new developments in large-scale NLP.

Large scale optimization problems with continuous variables and nonlinear constraints pose a number of challenges to current nonlinear programming algorithms. Unlike algorithms for discrete variables, NLP algorithms are necessarily iterative and do not have finite termination properties. Progress is therefore measured by *convergence rates* in the neighborhood of the (local) solution. Moreover, these algorithms are often more difficult to construct, analyze and make reliable and efficient for different problem types. Research in the development of efficient and reliable nonlinear programming solvers addresses the following areas:

Inequality constraint handling in large scale problems has typically faced a major combinatorial challenge for commercial methods. The recent introduction of interior point (or barrier) methods for NLP: KNITRO (Byrd et al., 2000), LOQO (Vanderbei and Shanno, 1999) and IPOPT (Waechter and Biegler, 2003) have shown significant improvements over conventional active set strategies. By replacing inequality constraints by barrier terms in the objective function and solving a sequence of rapidly converging NLPs, these approaches effectively remove combinatorial barriers for active set selection in large-scale problems.

Second order information for fast convergence. Most commercial solvers are limited in processing second derivatives and typically use quasi-Newton updates that are based on differences in first derivative information. This limits application problem sizes (e.g., <100 degrees of freedom) and algorithmic performance (often with linear convergence rates). Moreover, performance with quasi-Newton updates can be adversely affected by ill-conditioned problems. The availability of second derivatives in modern modeling environments (e.g., GAMS, AIMMS and AMPL) and in automatic differentiation tools (e.g., ADOL-C and Tapenade) overcomes these barriers but introduces new algorithmic challenges that include dealing with nonpositive curvature derived from second order information, and constructing efficient NLP methods that exploit sparsity of this information (Nocedal and Wright, 1999).

Globalization to find optimal solutions from poor starting points. Most commercial NLP strategies have limited capabilities to deal with poor initializations. This is a major barrier to incorporating them within design and operations tools. Over the past decade globalization strategies have been conceived that guarantee convergence to locally optimal solutions under mild conditions (Nocedal and Wright, 1999). Classified as *line search* and *trust region* methods they offer a number of trade-offs; line search methods are faster and easier to implement than trust regions, but the latter are more powerful in handling ill-conditioning and negative curvature. More recently convergence metrics for both methods improved with the introduction of *filter* methods (Fletcher et al., 2003; Waechter and Biegler, 2003), which rapidly eliminate undesirable search regions and promote global convergence.

Finally, *decomposition strategies* need to be developed that deal with large, structured problems, the incorporation of both equation-based and procedural models and the integration of models that occur at different spatial (lumped or distributed parameter systems) and temporal levels (e.g., for dynamic and static operation). Support for this activity is required in the development of optimization software that exploits parallel computing as well as flexible configuration and modification of algorithms tailored to exploit features of these models (Grossmann and Biegler, 2003).

To address and overcome some of these issues in nonlinear programming, we consider three case studies in this paper that *cannot be addressed with off-the-shelf optimization strategies*. Instead, we describe the development of powerful large-scale algorithms that exploit challenging features of the problem. In particular, we focus on the barrier NLP approach and describe its characteristics.

Without loss of generality, the optimization problem (NLP) can be stated as:

$$\begin{aligned} &Min\ f(x) \\ &s.t.\ c(x) = 0 \\ &x \geq 0 \end{aligned} \qquad (1)$$

We assume the objective function $f(x)$: $R^n \rightarrow R$ and the equality constraints $c(x)$: $R^n \rightarrow R^m$ with $m < n$ are sufficiently smooth. The bounds are now replaced by a logarithmic barrier term, which is added to the objective term to give:

$$\begin{aligned} Min\ \varphi(x) &= f(x) - \mu_\ell \sum_i \log(x^{(i)}) \\ s.t.\ \ c(x) &= 0 \end{aligned} \qquad (2)$$

The barrier method solves a sequence of barrier problems (indexed by ℓ) for decreasing values of μ_ℓ with $\lim_{\ell\to\infty} \mu_\ell = 0$. Under mild assumptions it can be shown that a sequence of $x^*(\mu_\ell)$ of (approximate) local solutions of (2) converges to a local solution of the original NLP (1) (Fiacco and McCormick, 1968; Forsgren et al., 2002). Since the exact solution $x^*(\mu_\ell)$ is not of interest for large values of μ_ℓ, the corresponding barrier problem is solved only to a relaxed accuracy ε_ℓ with $\lim_{\ell\to\infty} \varepsilon_\ell = 0$. To solve the barrier problem for a fixed value of μ_ℓ, a primal-dual approach is used to generate search directions for the primal and dual variables.

For fixed values of μ_ℓ, the barrier problem (2) is solved using a Newton method, with directions determined by solving the linear system at iteration k:

$$\begin{bmatrix} H(x_k,\lambda_k)+\Sigma_k & A(x_k) \\ A(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k \end{bmatrix} = -\begin{bmatrix} \nabla\varphi_\mu(x_k) \\ c(x_k) \end{bmatrix} \qquad (3)$$

where λ_k is the vector of multiplier estimates at iteration k, $A(x_k) = \nabla c(x_k)$, $H(x_k, \lambda_k)$ is the Hessian of the Lagrange function of (2) given by $L_\mu(x, \lambda) = f(x) + c(x)^T \lambda$, the diagonal matrix $\Sigma_k = X^{-1}V$ represents the barrier term and $X = diag(x)$ and $V = diag(v)$, with v the multiplier on the nonnegativity constraint in (1). Once the search direction is computed, a globalization method is applied to ensure convergence from poor starting points. Suitable methods include the filter line search method in IPOPT (Waechter and Biegler, 2003), the trust region approach in KNITRO (Byrd et al., 2000) or a combined approach in KNITRO-Direct (Walz et al., 2003), which all deal with steps generated by (3).

Solution of (3) can be done with either a decomposition into a reduced space, a direct solve in the full space, or an iterative solution based on preconditioned Krylov methods (Biros and Ghattas, 1990). Reduced space decompositions have been addressed in Cervantes et al. (1999) and are best suited for problems with few decision variables (i.e., degrees of freedom). In this case, second derivatives need not be calculated and curvature information in the reduced space of the decision variables can be approximated efficiently with quasi-Newton updates. Problems that benefit from this approach include applications in equipment and process design and parameter estimation. However, for many degrees of freedom this approach can become slow to converge and special attention must be paid to the costs of factorization.

On the other hand, the full space direct solution of (3) requires calculation of second derivatives for $H(x, \lambda)$ as well as an efficient large-scale sparse matrix solver. Advocates of this approach include Lucia and coworkers (1990, 1993), Betts and Frank (1995) and Sargent and Ding (2000). Moreover, with the application of barrier methods, algorithms can be constructed that easily take advantage of the structure of A_k and H_k in the KKT matrix without the additional expense of changing the active set. In addition, a second concern is due to an incorrect inertia (distribution of positive, negative and zero eigenvalues) in the KKT matrix. This results from linearly dependent columns in the A_k matrix or indefiniteness in the reduced Hessian. When encountered, they can be corrected by the following simple device:

$$\begin{bmatrix} H(x_k,\lambda_k)+\Sigma_k+\delta_1 I & A(x_k) \\ A(x_k)^T & -\delta_2 I \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k \end{bmatrix} = -\begin{bmatrix} \nabla\varphi_\mu(x_k) \\ c(x_k) \end{bmatrix} \qquad (4)$$

In both LOQO and IPOPT, δ_1 and δ_2 are systematically increased until the correct inertia (determined by the sparse linear solver) is found. This device is effective in treating degenerate features of the NLP problem. Nevertheless, slow convergence and also expensive linear factorizations of the KKT matrix may still be encountered.

The advantage of the full space approach arises when applied to problems with many degrees of freedom. These include NLPs resulting from the discretization of differential equations such as optimal control problems (to determine an accurate profile in time), state estimation and inverse problems. In process engineering, they also include large-scale problems in data reconciliation, blending operations, and model predictive control.

Data Reconciliation Example

To explore the performance of NLP solvers on process problems with many degrees of freedom, Poku et al. (2004) considered a set of blending and data reconciliation problems and compared several popular NLP algorithms. Here it was determined that the capability of including second derivative information and factorizing in the full space can lead to significant performance differences. In particular, consider the data reconciliation problem for a steam metering problem first introduced by Serth and Heenan (1986) and modified to include both flow and temperature measurements and bilinear constraints in Arora and Biegler (2001). The steam metering process has 28 redundant measured streams flowing in and out of 11 nodes. To increase the degrees of freedom, data snapshots were generated for one to 25 days. For this problem the number of variables increases from 40 to 976 and the degrees of freedom (also known as

superbasic variables) increase from 17 to 425. The comparison by Poku et al. (2004) is summarized in Figure 1.

Note that the iteration count for MINOS and SNOPT, methods, which do not use second order information and rely on quasi-Newton updates, increases linearly with the number of degrees of freedom; these are among the slowest NLP solvers. On the other hand, both KNITRO and IPOPT are quite fast and the number of iterations remains small as the problem size increases. As a result, problems with many degrees of freedom can be solved much more efficiently with these advanced solvers. Poku et al. (2004) also address a set of blending problems which are not as well behaved due to dependent constraints. Here the device in (4) allows the successful solution of these problems with IPOPT, even when the other methods failed.

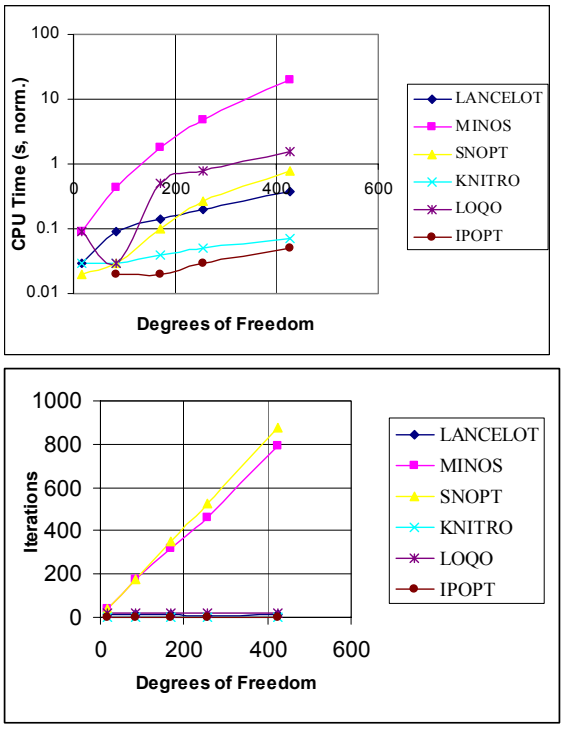

Figure 1. Comparison of NLP Solvers on Data Reconciliation Problem

Loss of Positive Curvature

With the availability of exact second derivatives, care must also be taken with the loss of positive curvature in the tangential space of the active constraints. If this occurs at isolated points along the convergence path, this indefiniteness can be corrected by adjustment of δ_1 in (3), or more rigorously through the application of a trust region method (Byrd et al., 2001). On the other hand, if positive curvature is absent at the solution, a sensible approach is to regularize the objective function through the addition of quadratic terms. This approach is efficient for line search algorithms in large problems but can lead to a *smeared* solution if nonunique solutions are present. Smeared solutions can be avoided if additional problem specific data is known and a tailored regularization can be designed. To demonstrate this approach on a large-scale problem we consider the detection of contaminants in a municipal water network.

Source Detection Example

To illustrate the importance of regularization and the formulation and solution of large-scale PDE problems, we consider the problem of detecting contaminant injections in municipal water networks. While this problem is not new, heightened awareness of such malicious attacks has caused renewed interest in this problem. Here we consider an optimization formulation for source detection that uses large-scale water network models. Models of municipal water networks are constructed by partial differential equations for network pipes that are a function of both time and pipe displacement. Mixing occurs at pipe nodes (junctions and mixing tanks) and any of these can function as an injection point. To detect potential contaminants, concentration sensors are distributed throughout the network. This leads to a large-scale dynamic optimization problem, where profiles of injected mass, $m_k(t)$, need to be detected at node k from sensor measurements. A typical network model is given below:

$$\left.\begin{aligned} &\frac{\partial c_i(x,t)}{\partial t} + u_i(t)\frac{\partial c_i(x,t)}{\partial x} = 0 \\ &c(x = I_i(t), t) = \hat{c}_{ki(t)}(t) \\ &c(x, t=0) = 0 \end{aligned}\right\} i \in P$$

$$\hat{c}_k(t) = \frac{\left(\sum_{i \in \Gamma_k(t)} Q_i(t) c_i(x = O_i(t), t)\right) + m_k(t)}{\left(\sum_{i \in \Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)} \qquad k \in J \qquad (5)$$

$$V_k \frac{d\hat{c}_k}{dt} = \left(\sum_{i \in \Gamma_k(t)} Q_i(t) c_i(x = O_i(t), t)\right) + m_k(t) - \left(\sum_{i \in \Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)$$

$$\hat{c}_k(t=0) = 0, k \in S$$

where the sets P, J, S are the sets of all pipes, junctions, and storage tanks, $0 \le t \le t_f$, and $x \ge 0$ is displacement along a pipe. Also, $u_i(t)$ is the known fluid velocity in pipe i, $c_i(x,t)$ is the contaminant concentration in pipe i, $\hat{c}_{ki(t)}(t)$ is the contaminant concentration of node k, $\Gamma_{k(t)}$ is the set of all pipes flowing into node k at time t, $I_{i(t)}$, and $O_{i(t)}$, are the displacements along pipe i where fluid is entering and leaving the pipe, respectively, and $k_{i(t)}$ is the the index of the node connected at the inlet of pipe i. Note that these designations are time dependent and change with the flow direction.

Since hydraulic calculations are known in advance, water quality calculations are decoupled from the flowrates and can be determined by a variety of existing techniques. A naïve discretization of this system in time and space produces a large scale, nonlinear math programming problem that is unreasonable for current optimization tools. To overcome this difficulty, we use a *Lagrangian* reference frame. Here, an origin tracking algorithm is developed in Laird et al. (2004) that reformulates the partial differential pipe expressions into algebraic equations with variable time delays, thus removing the need to discretize along the length of the pipes.

The resulting model becomes a set of DAEs with variable time delays that tracks the node concentrations, $\hat{c}_{ki(t)}(t)$, over time. The optimization is formulated from these DAEs and by adding the objective function and constraints, we obtain:

$$\begin{aligned} Min \quad & \sum_{r\in\Theta} \sum_{k} \int w_k(t)(\hat{c}_k(t) - \hat{c}_k^*(t))^2 \delta(t-t_r) + \rho\, m_k(t)^2\, dt \\ s.t. \quad & DAE\ Model\ (derived\ from\ (5)) \\ & m_k(t) \geq 0 \end{aligned} \tag{6}$$

where $\hat{c}_k^*(t)$ are the available measurements and Θ is the set that defines the time series of sampled data. Note that this problem is regularized with a parameter $\rho > 0$ to enforce positive curvature in the Jacobian and lead to a unique solution. Such a regularization does not prevent smeared detections if too few sensors are present, but it does lead to an efficient and reliable solution strategy. For our examples, we have found that a value of $\rho = 10^{-4}$ is sufficient.

To demonstrate this approach we consider the municipal water network shown in Figure 2. After discretizing this DAE model in time, we apply the IPOPT solver. Upon discretization the resulting NLP is a large problem with many degrees of freedom, with about 210,000 variables, 165,000 equality constraints and 45,000 inequalities. A typical NLP solution with IPOPT takes less than 2 CPU minutes on a 2.2 GHz Pentium 4 machine.

In Laird et al. (2004) we present the results of over 1000 tests with varying numbers of network sensors at injection points and different time dependent injection scenarios at points A, B, C and D in the network. These tests are made with data sets of up to 8 hours.

The formulation is also very effective for locations A through C, where it determines the injection location in very little time with few sensors. The effectiveness of the sensor grid is influenced dramatically by the flow patterns at the injection location. The effectiveness of the formulation is poorer for location D, because flow conditions at this point do not cause significant spreading of the contaminant through the network, and a higher requirement on the number of installed sensors is expected. Nevertheless, once there are enough sensors to detect the contamination, the formulation is extremely effective at determining the correct injection location.

Figure 2. Municipal network for source detection with trial injection points A-D.

This detection scheme is extremely fast and demonstrates the performance of efficient problem formulations coupled to specialized large-scale NLP solvers. Future work deals with augmenting the NLP problem formulation with network identification tools, similar to those for data reconciliation (see, e.g., Narasimhan et al., 2000). Optimal sensor location and problem dependent regularizations to sharpen the detection results will also be investigated.

Mathematical Programs with Complementarity Constraints (MPCCs)

In addition to using integer decision variables in the disjunctive problem formulations discussed in the previous sections, many discrete decisions can be modeled through continuous complementarity relations, e.g., $w^{(i)}y^{(i)} = 0,\ w,\ y \geq 0$. These have recently been considered in modeling dynamic hybrid systems (van der Schaft and Schumacher, 1998; Heemels et al., 2001) and in modeling disjunctions (Stein et al., 2004). Introducing complementarity constraints leads to a specialized nonlinear programming formulation that requires careful treatment in algorithmic design. It should also be noted that the introduction of complementarity conditions does introduce additional nonconvexity into the problem, which would not be observed, for instance, in the NLP subproblems of a mixed integer programming approach.

Mathematical programs with complementarity conditions (MPCCs) can be stated as:

$$\begin{aligned} &Min\ f(x, w, y) \\ &s.t\ \ c(x, w, y) = 0 \\ &\quad W y \le 0,\ w, y \ge 0,\ x \ge 0 \end{aligned} \qquad (7)$$

where $W = diag(w)$. The addition of these constraints leads to an NLP that does not satisfy the usual constraint qualifications (LICQ or MFCQ) and therefore has solution sets with unbounded multipliers. For this reason, many NLP codes have difficulties with MPCC formulations, although Fletcher and Leyffer (2004) and Fletcher et al. (2002) show that well implemented NLP codes can be used to solve some classes of MPCCs. Moreover, barrier methods such as IPOPT can be safeguarded and extended naturally to deal with MPCCs. Raghunathan and Biegler (2003) analyzed the structure of MPCCs and developed a tailored barrier method.

The MPCC can be relaxed to the following form (Clark and Westerberg, 1990; Luo et al., 1996):

$$\begin{aligned} &Min\ f(x, w, y) \\ &s.t.\ \ c(x, w, y) = 0 \\ &\quad W y + s = t,\ w, y \ge 0,\ x \ge 0,\ s \ge 0 \end{aligned} \qquad (8)$$

for some positive value for t. Scholtes (2001) showed that minor relaxations in (8) lead to well-defined parametric programs in t. Now replacing the inequalities with barrier terms leads to:

$$\begin{aligned} Min\ \ & f(x, w, y) - \mu_\ell \sum_i \log(x^{(i)}) - \mu_\ell \sum_i \log(w^{(i)}) \\ & - \mu_\ell \sum_i \log(y^{(i)}) - \mu_\ell \sum_i \log(s^{(i)}) \qquad (9) \\ s.t.\ \ & c(x, w, y) = 0 \\ & Wy + s = \delta\mu_\ell \end{aligned}$$

where δ is a positive constant. Raghunathan and Biegler (2003) showed that with a suitable regularization of the KKT matrix, this approach leads to a barrier method with $\mu_\ell \to 0$ that is superlinearly convergent. The resulting MPCC code, called IPOPT-C, has been compared on hundreds of test problems, including challenging applications in steady state distillation optimization with disappearing phases, optimal control and optimal startup of distillation processes with phase transitions and data reconciliation and parameter estimation of fermentation processes (Raghunathan et al., 2003a,b, 2004a, b). The ability to formulate and handle complementarity constraints in a reliable manner leads to difficult applications that are often too large to be considered with integer variables. This is illustrated below with the optimization of a hybrid dynamic system of a metabolic flux network for yeast fermentation.

Yeast Fermentation Example

Mathematical modeling and optimization of wine fermentation presents a major challenge as yeast strains must adapt to highly variable environmental conditions. Sainz et al. (2003) proposed a general framework for simulating the evolution of cell metabolism in a changing medium, based on metabolic flux balance models. Here the cell metabolism is represented by a linear program (LP) that maximizes physiological objectives such as optimal growth and homeostasis, subject to the network of metabolic reactions. However, the bounds on reaction rates in the LP and objective of the LP vary depending on concentrations of specific metabolites, which lead to cell adaptation to changing environment. Bounds on reaction rates are obtained based on experiments and are piecewise smooth functions of extracellular metabolite concentrations. The dynamic evolution of extracellular metabolites is then defined using Differential Algebraic Equations (DAEs) coupled with these LP-based metabolic models. Here our task is to estimate time-invariant parameters such as biomass compositions in order to match model predictions to experimental measurements of glucose and biomass concentrations in the medium.

The parameter estimation problem definition begins with the LP formulation based on the metabolic network shown in Figure 3. Assuming that environmental dynamics are much slower than enzymatic regulation, cell metabolism can be represented by a metabolic network in which all reactions considered are in pseudo-steady state and can be represented through linear mass balances and the following feasible region:

$$\Psi = \{v | A(\theta)\, v = 0,\ v^U(C) \ge v \ge v^L(C)\}$$

where v is the vector of metabolic rates, θ is the vector of parameters, such as biomass stoichiometric coefficients, to be estimated and C_i is the concentration of external metabolites ($i \in EXMET$). Note that the bounds on the rates are given as piecewise functions of these concentrations. The behavior of the metabolic flux network is represented by a linear program with feasible region Ψ and an objective to maximize biomass above a specified ammonium concentration and to minimize ATP consumption below this ammonium concentration. The parameter estimation problem is then given by:

$$\begin{aligned} &Min \sum_{j \in MEAS} \sum_{\tau \in T_j} (C_j(\tau) - C_j^{meas}(\tau))^2 \\ &s.t.\ \ \frac{dC_i}{dt} = v_i C_{bio}(t), C_i(0) = C_{i0} \quad i \in EXMET \qquad (10) \\ &\quad \min\ \ v^T d_1, s.t.\ \ v \in \Psi_1(v^L(C), v^U(C), \theta),\ \ C_{NH+} \ge \varepsilon \\ &\quad \min\ \ v^T d_2, s.t.\ \ v \in \Psi_2(v^L(C), v^U(C), \theta),\ \ C_{NH+} < \varepsilon \end{aligned}$$

where T_j is the set of time points for measurement j.

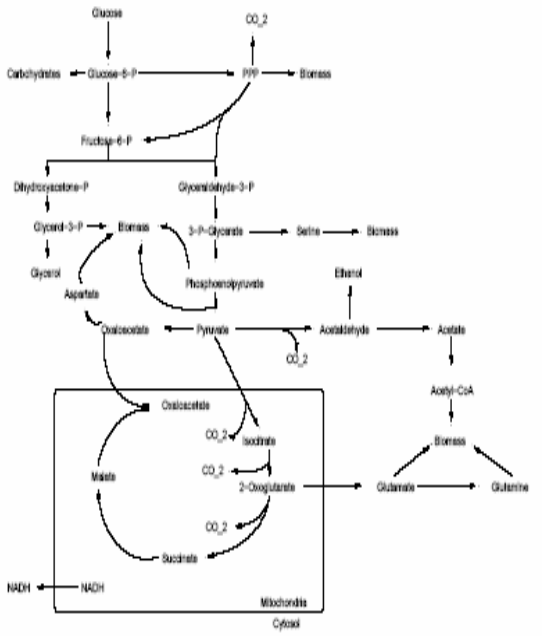

Figure 3. Metabolic Network for Yeast Fermentation

To solve this nonsmooth parameter estimation problem, Raghunathan et al. (2004) propose a novel formulation of (10) using Variational Inequalities (VIs) to represent the piecewise continuous bounds and multiple linear programs in a well-posed manner. Including the ODEs governing metabolite evolution in (10) along with the resulting VIs leads to a system of Differential Variational Inequalities (DVIs) governing fermentation dynamics.

This system is discretized using the trapezoidal rule to form a large-scale mathematical program with variational inequalities, which can be reformulated to form an MPCC problem. Here the resulting problem with VI constraints is given by:

$$\begin{aligned} &Min\ f(x, w) \\ &s.t.\ \ h(x, w) = 0,\ \ x \geq 0 \\ &\quad (\tilde{w}\text{-}w)^T F(x) \geq 0 \quad w, \forall \tilde{w} \in \Psi \end{aligned} \tag{11}$$

From $\tilde{w}^T F(x) \geq w^T F(x) \quad w, \forall \tilde{w} \in \Psi$ it is easy to see that w is given by the solution of $Min\ w^T F(x),\ w \in \Psi$ and (11) is given by the bi-level problem:

$$\begin{aligned} &Min\ f(x, w) \\ &s.t.\ \ h(x, w) = 0,\ \ x \geq 0 \\ &\quad Min\ w^T F(x), w \in \Psi \end{aligned} \tag{12}$$

Replacing the inner minimization problem with the corresponding KKT conditions leads to the following MPCC problem:

$$\begin{aligned} &Min\ f(x, w) \\ &s.t.\ \ h(x, w) = 0,\ \ x \geq 0 \\ &\quad c(x, w, y) = 0,\ \ w, y \geq 0 \\ &\quad W y \leq 0 \end{aligned} \tag{13}$$

Raghunathan et al. (2004) considered a number of nitrogen-rich and nitrogen-lean cases for this parameter estimation problem, using both simulated and laboratory data. An interesting problem that arises is the adequacy of the measurements to describe these networks. In particular they were able to:

- Demonstrate efficient algorithmic performance with sufficient measured data using simulation
- Demonstrate the adequacy of partial information in predicting unmeasured profiles
- Validate metabolic models with laboratory data for metabolite and biomass evolution.

Finally, the MPCC (13) that results from this hybrid dynamic parameter estimation problem has up to 37,328 variables and 29,583 constraints, including 7740 complementarity constraints. Note that in addition to handling large numbers of complementarity constraints, which model the equivalent number of discrete decisions, the MPCC also has many degrees of freedom that need to be handled efficiently. Solved with IPOPT-C, this MPCC model requires less that 8 CPU minutes on a 2.2 MHz Pentium 5 machine.

Logic-Based Discrete and Continuous Optimization

The mathematical programming approach to discrete/continuous optimization problems has been widely used in operations research and engineering. For example, the applications are in process design and synthesis, planning and scheduling, process control, and recently, in molecular design and in bioinformatics. Over the last decades, there has been a significant progress in the development of the discrete/continuous optimization models and their solution algorithms. For a recent review in the applications to the process systems engineering, see Grossmann et al. (1999). In this section of the paper we present an overview of the advances in mathematical programming for the modeling and solution of discrete/continuous optimization problems with special emphasis on logic-based optimization.

Review of Mixed Integer Optimization

The conventional way of modeling discrete/continuous optimization problems has been through the use of 0-1 and continuous variables, and algebraic equations and inequalities. For the case of linear functions this model corresponds to a mixed-integer linear programming (MILP) model, which has the following general form,

$$\begin{aligned} &\min Z = a^T y + b^T x \\ &s.t. \quad Ay + Bx \leq d \qquad\qquad (MILP) \\ &x \in R^n, y \in \{0,1\}^m \end{aligned}$$

In problem (MILP) the variables x are continuous, and y are discrete variables, which generally are binary variables. As is well known, problem (MILP) is NP-hard. Nevertheless, an interesting theoretical result is that it is possible to transform it into an LP with the convexification procedures proposed by Lovacz and Schrijver (1991), Sherali and Adams (1990), and Balas et al (1993). These procedures consist in sequentially lifting the original relaxed x-y space into higher dimension and projecting it back to the original space so as to yield after a finite number of steps the integer convex hull. Since the transformations have exponential complexity, they are only of theoretical interest, although they can be used as a basis for deriving cutting planes (e.g. lift and project method by Balas et al, 1993).

As for the solution of problem (MILP), it should be noted that this problem becomes an LP problem when the binary variables are relaxed as continuous variables, $0 \leq y \leq 1$. The most common solution algorithms for problem (MILP) are LP-based branch and bound methods, which are enumeration methods that solve LP subproblems at each node of the search tree. This technique was initially conceived by Land and Doig (1960), Balas (1965), and later formalized by Dakin, (1965). Cutting plane techniques, which were initially proposed by Gomory (1958), and consist of successively generating valid inequalities that are added to the relaxed LP, have received renewed interest through the works of Crowder et al (1983), Van Roy and Wolsey (1986), and especially the lift and project method of Balas et al (1993). A recent review of branch and cut methods can be found in Johnson et al. (2000). Finally, Benders decomposition (Benders, 1962) is another technique for solving MILPs in which the problem is successively decomposed into LP subproblems for fixed 0-1 and a master problem for updating the binary variables.

Software for MILP solver includes OSL, CPLEX and XPRESS which use the LP-based branch and bound algorithm combined with cutting plane techniques. MILP models and solution algorithms have been developed and applied successfully to many industrial problems (e.g. see Kallrath, 2000).

For the case of nonlinear functions the discrete/continuous optimization problem is given by Mixed-integer nonlinear programming (MINLP) model:

$$\begin{aligned} &\min Z = f(x, y) \\ &s.t. \quad g(x, y) \leq 0 \qquad\qquad (MINLP) \\ &x \in X, y \in Y \\ &X = \{x \mid x \in R^n, x^L \leq x \leq x^U, Bx \leq b\} \\ &Y = \{y \mid y \in \{0,1\}^m, Ay \leq a\} \end{aligned}$$

where $f(x,y)$ and $g(x,y)$ are assumed to be convex, differentiable and bounded over X and Y. The set X is generally assumed to be a compact convex set, and the discrete set Y is a polyhedral of integer points. Usually, in most applications it is assumed that $f(x,y)$ and $g(x,y)$ are linear in the binary variables y.

A recent review of MINLP solution algorithms can be found in Grossmann (2002). Algorithms for the solution of problem (MINLP) include the Branch and Bound (BB) method, which is a direct extension of the linear case of MILPs (Gupta and Ravindran, 1985; Borchers and Mitchell, 1994; Leyffer, 2001). The Branch-and-cut method by Stubbs and Mehrotra (1999), which corresponds to a generalization of the lift and project cuts by Balas et al (1993), adds cutting planes to the NLP subproblems in the search tree. Generalized Benders Decomposition (GBD) (Geoffrion, 1972) is an extension of Benders decomposition and consists of solving an alternating sequence of NLP (fixed binary variables) and aggregated MILP master problems that yield lower bounds. The Outer-Approximation (OA) method (Duran and Grossmann, 1986; Yuan et al., 1988; Fletcher and Leyffer, 1994) also consists of solving NLP subproblems and MILP master problems. However, OA uses accumulated function linearizations which act as linear supports for convex functions, and yield stronger lower bounds than GBD that uses accumulated Lagrangean functions that are parametric in the binary variables. The LP/NLP based branch and bound method by Quesada and Grossmann (1992) integrates LP and NLP subproblems of the OA method in one search tree, where the NLP subproblem is solved if a new integer solution is found and the linearization is added to the all the open nodes. Finally the Extended Cutting Plane (ECP) method by Westerlund and Pettersson (1995) is based on an extension of Kelley's cutting plane (1960) method for convex NLPs. The ECP method also solves successively an MILP master problem but it does not solve NLP subproblems as it simply adds successive linearizations at each iteration.

Generalized Disjunctive Programming

Given difficulties in the modeling and scaling of mixed-integer problems, the following major approaches based on logic-based techniques have emerged: Generalized Disjunctive Programming (GDP) (Raman and Grossmann, 1994), Mixed Logic Linear Programming

(MLLP) (Hooker and Osorio, 1999), and Constraint Programming (CP) (Hentenryck, 1989) The motivations for these logic-based modeling has been to facilitate the modeling, reduce the combinatorial search effort, and improve the handling the nonlinearities. In this paper we will mostly concentrate on Generalized Disjunctive Programming. A general review of logic-based optimization can be found in Hooker (1999).

Generalized Disjunctive Programming (GDP) (Raman and Grossmann, 1994) is an extension of disjunctive programming (Balas, 1979) that provides an alternate way of modeling (MILP) and (MINLP) problems. The general formulation of a (GDP) is as follows:

$$\begin{aligned} &\min Z = \sum_{k \in K} c_k + f(x) \\ &s.t. \quad g(x) \le 0 \\ &\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ h_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{bmatrix}, \ k \in K \\ &\Omega(Y) = True \\ &x \in R^n, c \in R^m, Y \in \{true, false\}^m \end{aligned} \qquad \text{(GDP)}$$

where Y_{jk} are the Boolean variables that decide whether a term j in a disjunction $k \in K$ is true or false, and x are continuous variables. The objective function involves the term $f(x)$ for the continuous variables and the charges c_k that depend on the discrete choices in each disjunction $k \in K$. The constraints $g(x) \le 0$ hold regardless of the discrete choice, and $h_{jk}(x) \le 0$ are conditional constraints that hold when Y_{jk} is true in the j-th term of the k-th disjunction. The cost variables c_k correspond to the fixed charges, and are equal to γ_{jk} if the Boolean variable Y_{jk} is true. $\Omega(Y)$ are logical relations for the Boolean variables expressed as propositional logic.

It should be noted that problem (GDP) can be reformulated as an MINLP problem by replacing the Boolean variables by binary variables y_{jk},

$$\begin{aligned} &min\, Z = \sum_{k \in K} \sum_{j \in J_k} \gamma_{jk} y_{jk} + f(x) \\ &s.t. \quad g(x) \le 0 \\ &h_{jk}(x) \le M_{jk}(1 - y_{jk}), \ j \in J_k, k \in K \qquad (BM) \\ &\sum_{j \in J_k} y_{jk} = 1, \ k \in K \\ &Ay \le a \\ &0 \le x \le x^U, \ y_{jk} \in \{0,1\}, \ j \in J_k, k \in K \end{aligned}$$

where the disjunctions are replaced by "Big-M" constraints which involve a parameter M_{jk} and binary variables y_{jk}. The propositional logic statements $\Omega(Y)$ = True are replaced by the linear constraints $Ay \le a$ as described by Williams (1985) and Raman and Grossmann (1991). Here we assume that x is a non-negative variable with finite upper bound x^U. An important issue in model (BM) is how to specify a valid value for the Big-M parameter M_{jk}. If the value is too small, then feasible points may be cut off. If M_{jk} is too large, then the continuous relaxation might be too loose yielding poor lower bounds. Therefore, finding the smallest valid value for M_{jk} is the desired selection. For linear constraints, one can use the upper and lower bound of the variable x to calculate the maximum value of each constraint, which then can be used to calculate a valid value of M_{jk}. For nonlinear constraints one can in principle maximize each constraint over the feasible region, which is a non-trivial calculation.

Convex Hull Relaxation of Disjunction

Lee and Grossmann (2000) have derived the convex hull relaxation of problem (GDP). The basic idea is as follows. Consider a disjunction $k \in K$ that has convex constraints,

$$\begin{aligned} &\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ h_{jk}(x) \le 0 \\ c = \gamma_{jk} \end{bmatrix} \\ &0 \le x \le x^U, c \ge 0 \end{aligned} \qquad \text{(DP)}$$

where $h_{jk}(x)$ are assumed to be convex and bounded over x. The convex hull relaxation of disjunction (DP), (see Stubbs and Mehrotra, 1999), is given as follows:

$$\begin{aligned} &x = \sum_{j \in J_k} v^{jk}, \qquad c = \sum_{j \in J} \lambda_{jk} \gamma_{jk} \\ &0 \le v^{jk} \le \lambda_{jk} x^U{}_{jk}, \ j \in J_k \\ &\sum_{j \in J_k} \lambda_{jk} = 1, \ 0 \le \lambda_{jk} \le 1, \ j \in J_k \qquad \text{(CH)} \\ &\lambda_{jk} h_{jk}(v^{jk} / \lambda_{jk}) \le 0, \ j \in J_k \\ &x, c, v^{jk} \ge 0, \ j \in J_k \end{aligned}$$

where v^{jk} are disaggregated variables that are assigned to each term of the disjunction $k \in K$, and λ_{jk} are the weight factors that determine the feasibility of the disjunctive term. Note that when λ_{jk} is 1, then the j'th term in the k'th disjunction is enforced and the other terms are ignored. The constraints $\lambda_{jk} h_{jk}(v^{jk} / \lambda_{jk})$ are convex if $h_{jk}(x)$ is convex as discussed on p. 160 in Hiriart-Urruty and Lemaréchal (1993). A formal proof can be found in Stubbs and Mehrotra (1999). Note that the convex hull (CH) reduces to the result by Balas (1985) if the constraints are linear. Based on the convex hull relaxation (CH), Lee and Grossmann (2000) proposed the following convex relaxation program of (GDP).

$$
\begin{aligned}
&min\, Z^L = \sum_{k\in K}\sum_{j\in J_k}\gamma_{jk}\lambda_{jk} + f(x)\\
&s.t. \quad g(x) \leqslant 0\\
&x = \sum_{j\in J_k} v^{jk}, \quad \sum_{j\in J_k}\lambda_{jk} = 1, \quad k\in K \qquad \text{(CRP)}\\
&0 \leqslant v^{jk} \leqslant \lambda_{jk} x^{U}{}_{jk}, \; j\in J_k, \quad k\in K\\
&\lambda_{jk} h_{jk}(v^{jk}/\lambda_{jk}) \leqslant 0, \; j\in J_k, \quad k\in K\\
&A\lambda \leqslant a\\
&0 \leqslant x, v^{jk} \leqslant x^U, 0 \leqslant \lambda_{jk} \leqslant 1, \; j\in J_k, k\in K
\end{aligned}
$$

where U is a valid upper bound for x and v. For computational reasons, the nonlinear inequality is written as $\lambda_{jk} h_{jk}(v^{jk}/(\lambda_{jk}+\varepsilon)) \leq 0$ where ε is a small tolerance. This inequality remains convex if $h_{jk}(x)$ is a convex function. Note that the number of constraints and variables increases in (CRP) compared with problem (GDP). Problem (CRP) has a unique optimal solution and it yields a valid lower bound to the optimal solution of problem (GDP) (Lee and Grossmann, 2000). Problem (CRP) can also be regarded as a generalization of the relaxation proposed by Ceria and Soares (1999) for a special form of problem (GDP). Grossmann and Lee (2003) proved that problem (CRP) has the useful property that the lower bound is greater than or equal to the lower bound predicted from the relaxation of problem (BM).

Solution Algorithms for GDP

Branch and Bound

For the linear case of problem (GDP) Beaumont (1991) proposed a branch and bound method which directly branches on the constraints of the disjunctions where no logic constraints are involved. Also for the linear case Raman and Grossmann (1994) developed a branch and bound method which solves GDP problem in hybrid form, by exploiting the tight relaxation of the disjunctions and the tightness of the well-behaved mixed-integer constraints. There are also branch and bound methods for solving problem (GDP). In particular, a disjunctive branch and bound method can be developed that directly branches on the term in a disjunction using the convex hull relaxation (CRP) as a basic subproblem (Lee and Grossmann, 2000). Problem (CRP) is solved at the root node of the search tree. The branching rule is to select the least infeasible term in a disjunction first. Next, we consider a dichotomy where we fix the value $\lambda_{jk} = 1$ for the disjunctive term that is closest to being satisfied, and consider on the other hand the convex hull of the remaining terms ($\lambda_{jk} = 0$).

When all the decision variables λ_{jk} are fixed, problem (CRP) yields an upper bound to problem (GDP). The search is terminated when the lower and the upper bounds are the same. The algorithm has finite convergence since the number of the terms in the disjunction is finite. Also, since the nonlinear functions are convex, each subproblem has a unique optimal solution, and hence the the bounds are rigorous.

Reformulation and Cutting planes

Another approach for solving a linear GDP is to replace the disjunctions either by Big-M constraints or by the convex hull of each disjunction (Balas, 1985; Raman and Grossmann, 1994). For the nonlinear case a similar way for solving the problem (GDP) is to reformulate it into the MINLP by restricting the variables λ_{jk} in problem (CRP) to 0-1 values. Alternatively, to avoid introducing a potentially large number of variables and constraints, the GDP might also be reformulated as the MINLP problem (BM) by using Big-M parameters. One can then apply standard MINLP solution algorithms (i.e., branch and bound, OA, GBD, and ECP).

To strengthen the lower bounds one can derive cutting planes using the convex hull relaxation (CRP). To generate a cutting plane, the following 2-norm separation problem (SP), a convex NLP, is solved:

$$
\begin{aligned}
&\min \phi(x) = (x - x_R^{BM,n})^T (x - x_R^{BM,n})\\
&s.t. \quad g(x) \leqslant 0\\
&x = \sum_{i\in D_k} \nu_{ik} \quad , k\in K\\
&y_{ik} h_{ik}(\nu_{ik}/y_{ik}) \leqslant 0 \quad , i\in D_k, k\in K \qquad (SP)\\
&\sum_{i\in D_k} y_{ik} = 1 \quad , k\in K\\
&Ay \leqslant a\\
&x, \nu_{ik} \in R^n, 0 \leqslant y_{ik} \leqslant 1
\end{aligned}
$$

where $x_R^{BM,n}$ is the solution of problem (BM) with relaxed $0 \le y_{ik} \le 1$. Problem (SP) yields a solution point x^* which belongs to the convex hull of the disjunction and is closest to the relaxation solution $x_R^{BM,n}$. The most violated cutting plane is then given by,

$$(x^* - x_R^{BM,n})^T (x - x^*) \geq 0 \qquad \text{(CP1)}$$

The cutting plane in (CP1) is a valid inequality for problem (GDP). Problem (BM) is modified by adding the cutting plane (CP1) as follows:

$$\min Z = \sum_{k\in K}\sum_{i\in D_k}\gamma_{ik}\,y_{ik} + f(x)$$
$$s.t.\quad g(x) \leqslant 0$$
$$h_{ik}(x) \leqslant M_{ik}(1 - y_{ik}) \quad ,i\in D_k, k\in K \qquad \text{(CP)}$$
$$\sum_{i\in D_k} y_{ik} = 1 \quad ,k\in K$$
$$Ay \leqslant a$$
$$\beta^T x \leqslant b$$
$$x\in R^n, 0 \leqslant y_{ik} \leqslant 1$$

where $\beta^T x \le b$ is the cutting plane (CP1). Since we add a valid inequality to problem (BM), the lower bound obtained from problem (CP) is generally tighter than before adding the cutting plane.

This procedure for generating the cutting plane can be used by solving the separation problem (SP) only at the root node. It can also be used to strengthen the MINLP problem (BM) before applying methods such as OA, GBD, and ECP. It is also interesting to note that cutting planes can be derived in the (*x,y*) space, especially when the objective function has binary variables *y*.

Another application of the cutting plane is to determine if the convex hull formulation yields a good relaxation of a disjunction. If the value of $\| x^* - x_R^{BM,n} \|$ is large, then it is an indication that this is the case. A small difference between x^* and $x_R^{BM,n}$ would indicate that it might be better to simply use the Big-M relaxation. It should also be noted that Sawaya and Grossmann (2004) have recently developed the cutting plane method for linear GDP problems using the 1, 2 and ∞ norms, and relying on the theory of subgradient optimization.

GDP Decomposition Methods

Türkay and Grossmann (1996) have proposed logic-based OA and GBD algorithms for problem (GDP) by decomposition into NLP and MILP subproblems. For fixed values of the Boolean variables, Y_{jk} = true and Y_{ik} = false for $j \neq i$, the corresponding NLP subproblem is derived from (GDP) as follows:

$$\min Z = \sum_{k\in K} c_k + f(x)$$
$$s.t.\quad g(x) \le 0$$
$$\left.\begin{array}{l} h_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{array}\right\} \; for\ Y_{jk} = true, j\in J_k,\ k\in K \qquad \text{(NLPD)}$$
$$\left.\begin{array}{l} B^i x = 0 \\ c_k = 0 \end{array}\right\} \; for\ Y_{ik} = false, i\in J_k,\ k\in K$$
$$Ay \le a$$
$$x\in R^n, c\in R^m$$

For every disjunction *k* only the constraints corresponding to the Boolean variable Y_{jk} that is true are enforced. Also, fixed charges γ_{jk} are applied to these terms. After *K* subproblems (NLPD) are solved sets of linearizations l =1,...,*K* are generated for subsets of terms L_{jk} = { l | Yl_{jk} = *true* }, then one can define the following disjunctive OA master problem:

$$\min Z = \sum_{k\in K} c_k + \alpha$$
$$s.t.\quad \left.\begin{array}{l} \alpha \ge f(x^l) + \nabla f(x^l)^T (x - x^l) \\ g(x^l) + \nabla g(x^l)^T (x - x^l) \le 0 \end{array}\right\} \quad l = 1,\ldots,L$$
$$\left[\begin{array}{c} Y_{jk} \\ h_{jk}(x^l) + \nabla h_{jk}(x^l)^T (x - x^l) \le 0, l\in L_{jk} \\ c_k = \gamma_{jk} \end{array}\right] \vee \left[\begin{array}{c} \neg Y_{jk} \\ B^k x = 0 \\ c_k = 0 \end{array}\right], k\in K \quad \text{(MGPD)}$$
$$\Omega(Y) = True$$
$$x\in R^n, c\in R^m, Y\in \{true, false\}^m$$

Before solving the MILP master problem it is necessary to solve various subproblems (NLPD) in order to produce at least one linear approximation of each of the terms in the disjunctions. As shown by Türkay and Grossmann (1996) selecting the smallest number of subproblems amounts to the solution of a set covering problem. In the context of flowsheet synthesis problems, another way of generating the linearizations in (MGDP) is by starting with an initial flowsheet and optimizing the remaining subsystems as in the modeling/decomposition strategy (Kocis and Grossmann, 1987).

Problem (MGDP) can be solved by the methods described by Beaumont (1991), Raman and Grossmann (1994), and Hooker and Osorio (1999). For the case of process networks, Türkay and Grossmann (1996) have shown that if the convex hull representation of the disjunctions in (MGDP) is used, then assuming $B_k = I$ and converting the logic relations $\neg(Y)$ into the inequalities $Ay \le a$, leads to the MILP reformulation of (NLPD) which can be solved with OA. Türkay and Grossmann (1996) have also shown that while a logic-based Generalized Benders method (Geoffrion, 1972) cannot be derived as in the case of the OA algorithm, one can exploit the property for MINLP problems that performing one Benders iteration (Türkay and Grossmann, 1996) on the MILP master problem of the OA algorithm, is equivalent to generating a Generalized Benders cut. Therefore, a logic-based version of the Generalized Benders method performs one Benders iteration on the MILP master problem. Also, slack variables can be introduced to problem (MGDP) to reduce the effect of nonconvexity as in the augmented-penalty MILP master problem (Viswanathan and Grossmann, 1990).

Hybrid GDP/MINLP

Vecchietti and Grossmann (1999) have proposed a hybrid formulation of the GDP and algebraic MINLP

models. It involves disjunctions and mixed-integer constraints as follows:

$$
\begin{aligned}
&\min Z = \sum_{k \in K} c_k + f(x) + d^T y \\
&s.t.\ \ g(x) \le 0 \\
&\quad r(x) + Dy \le 0 \\
&\quad Ay \le a \\
&\bigvee_{j \in J_k} \begin{bmatrix} Y_{jk} \\ h_{jk}(x) \le 0 \\ c_k = \gamma_{jk} \end{bmatrix}, \ k \in K \qquad \text{(PH)} \\
&\quad \Omega(Y) = True \\
&x \in R^n, c \in R^m, y \in \{0,1\}^q, Y \in \{true, false\}^m
\end{aligned}
$$

where x and c are continuous variables and Y and y are discrete variables. Problem (PH) can reduce to a GDP or to an MINLP, depending on the absence and presence of the mixed-integer constraints and disjunctions and logic propositions. Thus, problem (PH) provides the flexibility of modeling an optimization problem as a GDP, MINLP or a hybrid model, making it possible to exploit the advantage of each model.

An extension of the logic-based OA algorithm for solving problem (PH) has been implemented in LOGMIP, a computer code based on GAMS (Vecchietti and Grossmann, 1999). This algorithm decomposes problem (PH) into two subproblems, the NLP and the MILP master problems. With fixed discrete variables, the NLP subproblem is solved. Then at the solution point of the NLP subproblem, the nonlinear constraints are linearized and the disjunction is relaxed by convex hull to build a master MILP subproblem which will yield a new discrete choice of (y,Y) for the next iteration.

Global Optimization Algorithm of Nonconvex GDP

In the above sections of the paper we assumed convexity in the nonlinear functions. However, in many applications nonlinearites give rise to nonconvex functions that may yield local solutions, not guaranteeing the global optimality. Global optimization of nonconvex programs has received increased attention due to their practical importance. Most of the deterministic global optimization algorithms are based on spatial branch and bound algorithm (Horst and Tuy, 1996), which divides the feasible region of continuous variables and compares lower bound and upper bound for fathoming each subregion. The one that contains the optimal solution is found by eliminating subregions that are proved not to contain the optimal solution.

For nonconvex NLP problems, Quesada and Grossmann (1995) proposed a spatial branch and bound algorithm for concave separable, linear fractional and bilinear programs using of linear and nonlinear underestimating functions (McCormick, 1976). For nonconvex MINLP, Ryoo and Sahinidis (1995) and later Tawarmalani and Sahinidis (2000) developed BARON, which branches on the continuous and discrete variables with bounds reduction method. Adjiman et al. (1997; 2000) proposed the SMIN-αBB and GMIN-αBB algorithms for twice-differentiable nonconvex MINLPs. Using a valid convex underestimation of general functions as well as for special functions, Adjiman et al. (1996) developed the αBB method which branches on both the continuous and discrete variables according to specific options. The branch-and-contract method (Zamora and Grossmann, 1999) has bilinear, linear fractional, and concave separable functions in the continuous variables and binary variables, uses bound contraction and applies the outer-approximation (OA) algorithm at each node of the tree. Kesavan and Barton (2000) developed a generalized branch-and-cut (GBC) algorithm, and showed that their earlier decomposition algorithm (Kesavan and Barton, 1999) is a specific instance of the GBC algorithm with a set of heuristics. Smith and Pantelides (1997) proposed a reformulation method combined with a spatial branch and bound algorithm for nonconvex MINLP and NLP, which is implemented in the gPROMS modeling system.

GDP Global Optimization Algorithms

We briefly describe two global optimization algorithms. The first was proposed by Lee and Grossmann (2001) and is for the case when the problem (GDP) involves bilinear, linear fractional and concave separable functions. First, these nonconvex functions of continuous variables are relaxed by replacing them with underestimating convex functions (McCormick, 1976; Quesada and Grossmann, 1995). Next, the convex hull of each nonlinear disjunction is constructed to build a convex NLP problem (CRP). At the first step, an upper bound is obtained by solving the nonconvex MINLP reformulation (BM) with the OA algorithm. This upper bound is then used for the bound contraction. The feasible region of continuous variables is contracted with an optimization subproblem that incorporates the valid underestimators and the upper bound value and that minimizes or maximizes each variable in turn. The tightened convex GDP problem is then solved in the first level of a two-level branch and bound algorithm, in which a discrete branch and bound search is performed on the disjunctions to predict lower bounds. In the second level, a spatial branch and bound method is used to solve nonconvex NLP problems for updating the upper bound. The algorithm exploits the convex hull relaxation for the discrete search, and the fact that the spatial branch and bound is restricted to fixed discrete variables in order to predict tight lower bounds.

The second algorithm is by Bergamini et al. (2004) that does not require spatial branch and bound searches as it uses piecewise linear approximations. The algorithm considers the Logic-Based Outer Approximation (OA) algorithm (Turkay and Grossmann, 1996) and is based on constructing a master problem that is a valid bounding representation of the original problem, and by solving the NLP subproblems to global optimality. The functions are assumed to be sums of convex, bilinear, and concave terms. To rigorously maintain the bounding properties of the MILP master problem, linear under and overestimators for bilinear, and concave terms are constructed over a grid with the property of having zero gap in the finite set of points. The set of these approximation points are defined over subdomains defined by bounds of variables and solution points of the previous NLP subproblems. For bilinear terms, the convex envelope by McCormick is used. Disjunctions are used to formulate the convex envelope in each subdomain, and the convex hull of these disjunctions is used to provide the tightest relaxation. It should be noted that binary variables are needed for the discrete choice of the corresponding subdomains. Linear fractional functions are treated similarly. Piecewise linear subestimations replace the concave terms.

The solution of the NLP subproblems to global optimality can be performed by fixing the topology variables in the MILP and by successively refining the grid of the piece-wise linear approximations. Alternatively, a general purpose NLP algorithm for global optimization (e.g. BARON code by Tawarmalani and Sahinidis, 2000) can be used. It should be noted that the NLP subproblems are reduced problems, involving only continuous variables related to a process with fixed structure. This allows the tightening of the variable bounds, and therefore reducing the computational cost of solving it to global optimality.

Constraint Programming and Hybrid MILP/CP Methods

In order to overcome difficulties in modeling and scalability of mathematical programming (MP) models, a trend that has emerged is to combine MP with symbolic logic reasoning into the quantitative. Among these attempts one of the more promising approaches has been the development of Constraint Programming (CP), which has proved to be particularly effective in scheduling applications. CP is essentially based on the idea that inference methods can accelerate the search for a solution.

Constraint Programming (CP) (van Hentenryck, 1989; Hooker, 2000) is a relatively new modeling and solution paradigm that was originally developed to solve feasibility problems, but it has been extended to solve optimization problems as well. Constraint Programming is very expressive as continuous, integer, as well as Boolean variables are permitted and moreover, variables can be indexed by other variables. Constraints can be expressed in algebraic form (e.g. $h(x) \leq 0$), as disjunctions (e.g. $[\ A1x \leq b1\ ] \vee [\ A2x \leq b2\ ]$), or as conditional logic statements (e.g. If $g(x) \leq 0$ then $r(x) \leq 0$). In addition, the language can support special implicit functions such as the *all different(x1, x2, ...xn)* constraint for assigning different values to the integer variables *x1, x2, ...xn.* The language consists of C++ procedures, although the recent trend has been to provide higher level languages such as OPL. Other commercial CP software packages include ILOG Solver (ILOG, 1999), CHIP (Dincbas *et al*., 1988), and ECLiPSe (Wallace *et al*., 1997).

Optimization problems in CP are solved as Constraint Satisfaction Problems (CSP), where we have a set of variables, a set of possible values for each variable (domain) and a set of constraints among the variables. The question to be answered is as follows: Is there an assignment of values to variables that satisfy all constraints? The solution of CP models is based on performing constraint propagation at each node by reducing the domains of the variables. If an empty domain is found the node is pruned. Branching is performed whenever a domain of an integer, binary or boolean variable has more than one element, or when the bounds of the domain of a continuous variable do not lie within a tolerance. Whenever a solution is found, or a domain of a variable is reduced, new constraints are added. The search terminates when no further nodes must be examined. The effectiveness of CP depends on the propagation mechanism behind constraints. Thus, even though many constructs and constraints are available, not all of them have efficient propagation mechanisms. For some problems, such as scheduling, propagation mechanisms have been proven to be very effective. Some of the most common propagation rules for scheduling are the "time-table" constraint (Le Pape, 1998), the "disjunctive-constraint" propagation (Baptiste and Le Pape, 1996; Smith and Cheng, 1993), the "edge-finding" (Nuijten, 1994; Caseau and Laburthe, 1994) and the "not-first, not-last" (Baptiste and Le Pape, 1996).

Since the two approaches appear to have complementary strengths, in order to solve difficult problems that are not effectively solved by either of the two, several researchers have proposed models that integrate the two paradigms. The integration between MILP and CP can be achieved in two ways (Hooker, 2002; van Hentenryck, 2002):

(a) By combining MILP and CP constraints into one hybrid model. In this case a hybrid algorithm that integrates constraint propagation with linear programming in a single search tree is also needed

for the solution of the model (e.g. see Heipcke et al., 1999; Rodosek, et al., 1999).

(b) By decomposing the original problem into two subproblems: one MILP and one CP subproblem. Each model is solved separately and information obtained while solving one subproblem is used for the solution of the other subproblem (Jain and Grossmann, 2001; Bockmayr and Pisaruk, 2003).

Maravelias and Grossmann (2004) have recently developed a hybrid MILP/CP method for the continuous time STN model and in which different objectives such as profit maximization, cost minimization and makespan minimization can be handled. The proposed method relies on an MILP model that represents an aggregate of the original MILP model. This method has shown to produce order of magnitude reductions in CPU times compared to standalone MILP or CP models.

Examples of logic-based optimization

Synthesis of Separation System

This problem was a joint collaboration with BP (Lee et al., 2003). It deals with the synthesis of a separation system of an ethylene plant in which a number of separation technologies such as dephlegmators, membranes, PSA, physical and chemical absorption, were considered in addition to the standard distillation columns and cold boxes. The superstructure of this problem is shown in Fig. 7, while the optimal solution is given in Fig 8. This problem was formulated as a GDP problem and reformulated as an MINLP by applying both big-M and convex hull transformations. The problem involved 5,800 0-1 variables, 24,500 continuous variables and 52,700 constraints, and was solved with GAMS DICOPT (CONOPT2/CPLEX) in 3 hours of CPU-time on a Pentium-III machine. Compared to the base-case design the optimal flowsheet included a dephlegmator and a physical absorber, and one less distillation column, achieving a $20 million reduction in the cost, largely from refrigeration.

Retrofit Planning Problem

In this problem it is assumed that an existing process network is given where each process can possibly be retrofitted for improvements such as higher yield, increased capacity, and reduced energy consumption. Given limited capital investments to make process improvements and cost estimations over a given time horizon, the problem consists of identifying those modifications that yield the highest economic improvement in terms of economic potential, which is defined as the income from product sales minus the cost of raw materials, energy and process modifications. Sawaya and Grossmann (2004) have developed a GDP model for this problem, which is a modification of work by Jackson and Grossmann (2002).

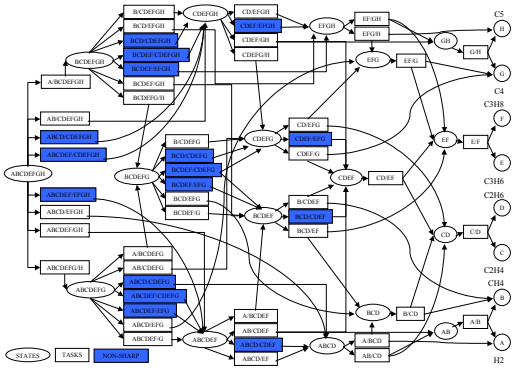

Figure 7: Superstructure of Separation of Ethylene Plant

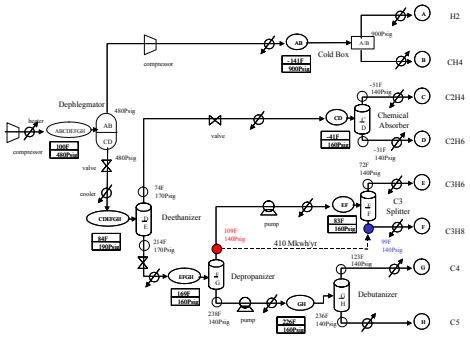

Figure 8:.Optimal Structure of Ethylene Plant

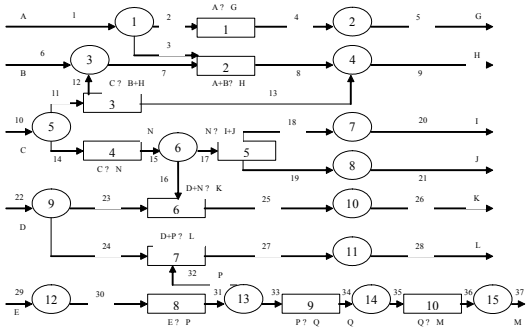

Fig. 9: Process network for retrofit planning

For a ten process instance (see Fig. 9) that involves the production of products (G,H,I,J,K,L,M) from raw materials (A,B,C,D,E), the 4-period MILP model was formulated with the big-M and convex hull reformulations. The former involved 320 0-1 variables, 377 continuous variables, and 1957 constraints**;** the latter involved 320 0-1 variables, 1097 continuous variables and 2505 constraints. The big-M model was solved in 1913 secs and 1,607,486 nodes, while the latter only required 5.8 secs and 2,155 nodes. This reduction was achieved because the convex hull formulation had a gap of only 7.6% versus the 60.3 gap of the big-M model. It should be noted that with 120

cuts the gap in the big-M model reduced to only 7.9%, with which the MILP was solved in a total of 68 secs, of which 22 were for the cut generation.

Wastewater treatment network

This example corresponds to a synthesis problem of a distributed wastewater multicomponent network, which is taken from example 10 of Galan and Grossmann (1998). Given a set of process liquid streams with known composition, a set of technologies for the removal of pollutants, and a set of mixers and splitters, the objective is to find the interconnections of the technologies and their flowrates to meet the specified discharge composition of pollutant at minimum total cost. Discrete choices involve deciding what equipment to use for each treatment unit. Fig. 10 shows the superstructure of a specific example with 3 contaminant and 3 choices of separation technologies per contaminant. Lee and Grossmann (2001) formulated the problem as a GDP model that involves 9 Boolean variables, 237 continuous variables and 281 constraints. The two level branch and bound method by Lee and Grossmann (2002) required about 5 minutes of CPU-time, while the method by Belgramini et al. (2004) required less than 2 minutes. The optimal solution with a cost of 1,692,583 $/year is shown in Fig. 11.

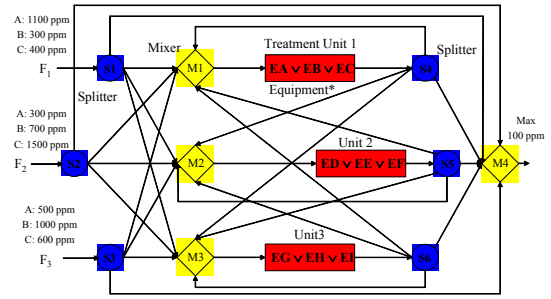

Figure 10:. Superstructure water treatment plant.

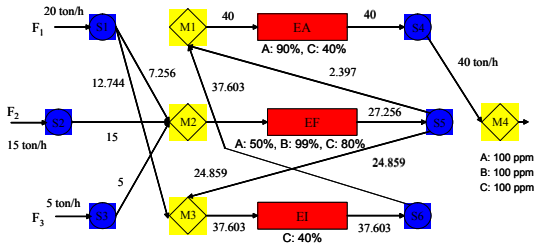

Figure 11: Optimal wastewater treatment plant.

Scheduling of Batch Plants

Consider the State-Task-Network shown in Fig. 12 that is an extension of the work by Papageorgiou and Pantelides (1996). The STN consists of 27 states, 19 tasks and has 8 equipment units available for the processing. The objective is to find a schedule that produces 5 tons each for products P1, P2, P3 and P4. The problem was originally modeled with the continuous-time MILP by Maravelias and Grossmann (2003) involving around 400 0-1 variables, 4,000 continuous variables and 6,000 constraints. Not even a feasible solution to this problem could be found with CPLEX 7.5 after 10 hours. In contrast, the proposed hybrid MILP/CP model required only 2 seconds and 5 major iterations between the MILP and CP subproblems! Note that the optimal schedule shown in Fig 13 is guaranteed to be the global optimum solution.

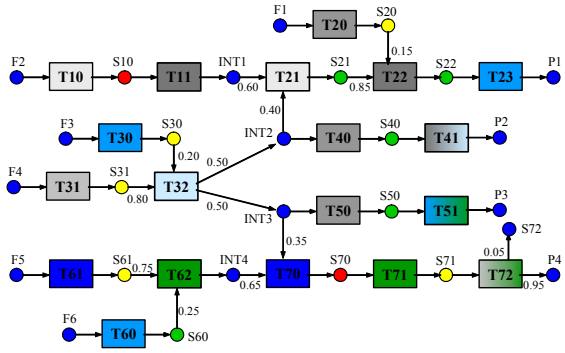

Fig. 12. State-Task-Network example.

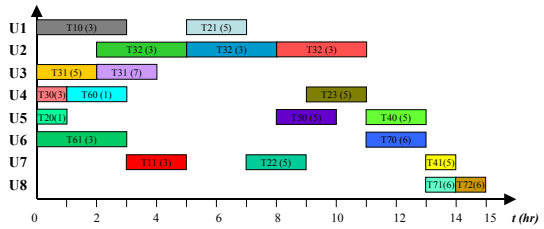

Fig. 13. Optimal schedule.

Conclusions

Nonlinear programming solvers are essential for optimization problems in product design and process design and operation. In addition to solving challenging problems in their own right, they also are essential algorithmic components within mixed integer programming and global optimization strategies. On the other hand, current off-the-shelf NLP solvers are often ill-prepared for large-scale challenges posed in design problems. These include the treatment of problems with many degrees of freedom, ill-posed large-scale problems that arise in the discretization of PDE and DAE models and the incorporation of complementarity conditions that can model certain classes of discrete decisions. To handle these challenges we illustrate the use of large-scale barrier methods as well as applications and refinements of the IPOPT algorithm. The efficiency and effectiveness of this approach is demonstrated on three NLP case studies (in data reconciliation, source detection and fermentation optimization) that cannot be handled efficiently with standard off-the-shelf NLP methods.

Mixed-integer optimization techniques (MILP, MINLP) have proved to be essential in modeling planning and

scheduling problems, as well as synthesis and design problems. The former tend to be largely linear, while the latter tend to be nonlinear. Major barriers that have been encountered with these techniques are modeling, scaling and nonconvexities. It is the first two issues that have motivated logic-based optimization as a way of facilitating the modeling of discrete/continuous problems, and of reducing the combinatorial search space. The GDP formulation has shown to be effective in terms of providing a qualitative/quantitative framework for modeling, and an approach that yields tighter relaxations through the convex hull formulation. It was also shown that global optimization algorithms can be developed for GDP models and solved in reasonable time for modest sized problem. Finally, the recent emergence of Constraint Programming (CP) offers an alternative approach for handling logic in discrete scheduling problems. Here the development of hybrid methods for scheduling seems to be particularly promising for achieving order of magnitude reductions in the computations.

Acknowledgments

The authors gratefully acknowledge financial support from the National Science Foundation under Grants ACI-0121497 and ACI-0121667 and from the industrial members of the Center for Advanced Process Decision-Making (CAPD) at Carnegie Mellon University.

References

Adjiman C.S., I.P. Androulakis, C.D. Maranas and C.A. Floudas, A Global Optimization Method, αBB, for Process Design. *Computers and Chem. Engng*., **20**, Suppl., S419-S424, 1996.

Adjiman C.S., I.P. Androulakis and C.A. Floudas, Global Optimization of MINLP Problems in Process Synthesis and Design. *Computers and Chem. Engng*., **21**, Suppl., S445-S450, 1997.

Adjiman C.S., I.P. Androulakis and C.A. Floudas, Global Optimization of Mixed-Integer Nonlinear Problems. *AIChE Journal*, **46**(9), 1769-1797, 2000.

Arora, N. and L. T. Biegler, "Redescending estimators for data reconciliation and parameter estimation," *Computers and Chemical Engineering,* 25, p. 1585 (2001)

Balas E., An Additive Algorithm for Solving Linear Programs with Zero-One Variables. *Operations Research*, **13**, 517-546, 1965.

Balas E., Disjunctive Programming. *Annals of Discrete Mathematics*, **5**, 3-51, 1979.

Balas E., Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM J. Alg. Disc. Meth.* 6. 466-486, 1985.

Balas E., S. Ceria and G. Cornuejols, A Lift-and-Project Cutting Plane Algorithm for Mixed 0-1 Programs. *Mathematical Programming*, **58**, 295-324, 1993.

Beaumont N., An Algorithm for Disjunctive Programs. *European Journal of Operations Research*, **48**, 362-371, 1991.

Benders J.F., Partitioning Procedures for Solving Mixed Variables Programming Problems. *Numerische Mathematik*, **4**, 238-252, 1962.

Betts, J. T., Frank, P. D., "A sparse nonlinear optimization algorithm," *J. Opt. Theo. Applics*., 3, pp. 519-541, 1994

Bisschop, J. and R. Entriken, *AIMMS The modeling system*. Paragon Decision Technology (1993)

Biros, G. and Omar Ghattas, "Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization." (2001) Submitted for publication.

Borchers B. and J.E. Mitchell, An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programming. *Computers and Operations Research*, **21**, 359-367, 1994.

Byrd, R. H. J. C. Gilbert and Jorge Nocedal, "An Interior Point Algorithm for Large Scale Nonlinear Programming," *Mathematical Programming*, 89, p. 149 (2000)

Ceria S. and J. Soares, Convex Programming for Disjunctive Optimization, *Mathematical Programming*, **86**(3), 595-614, 1999.

Cervantes, A. M., A. Waechter, R. Tutuncu and L. T. Biegler, "A Reduced Space Interior Point Strategy for Optimization of Differential Algebraic Systems," *Comp. Chem. Engr.* , 24, pp. 39-51 (2000)

Clark, P. A., and A. W. Westerberg, Bilevel programming for steady state chemical process design I, *Comp. and Chem. Eng*.,14(1), pp. 87-97 (1990).

Crowder H.P., E.L. Johnson and M.W. Padberg, Solving Large-Scale Zero-One Linear Programming Problems. *Operations Research*, **31**, 803-834, 1983.

Dakin R.J., A Tree Search Algorithm for Mixed Integer Programming Problems. *Computer Journal*, **8**, 250-255, 1965.

Dincbas, M., Van Hentenryck, P., Simonis, H., Aggoun, A., Graf, T. & Berthier, F. (1988). The constraint logic programming language CHIP. *In FGCS-88: Proceedings of International Conference on Fifth Generation Computer Systems, Tokyo*, 693-702.

Duran M.A. and I.E. Grossmann, An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Mathematical Programming,* **36**, 307-339, 1986.

Fiacco A. V., McCormick G. P., *Nonlinear Programming Sequential Unconstrained Minimization Techniques*, John Wiley, New York, 1990

Fletcher, R., S. Leyffer, D. Ralph, and S. Scholtes, "Local convergence of {SQP} methods for mathematical programs with equilibrium constraints." Technical Report NA 209, University of Dundee, UK, 2002.

Fletcher, R. and S. Leyffer, "Solving mathematical programs with complementarity constraints as nonlinear programs," *Optimization Methods and Software,* 18/1, pp. 15-40 (2004)

Fletcher R. and S. Leyffer, Solving Mixed Nonlinear Programs by Outer Approximation. *Mathematical Programming*, **66**(3), 327-349, 1994.

Forsgren A., Gill P. E. and M. H. Wright, "Interior Methods for Nonlinear Optimization," *SIAM Review*, 44, 4, pp. 525-597 2002

Galan, B. and I.E. Grossmann, "Optimal Design of Distributed Wastewater Treatment Networks," *Ind.Eng.Chem. Res.* 37, 4036-4048 (1998)

Geoffrion A.M., Generalized Benders Decomposition. *Journal of Optimization Theory and Application*, **10**(4), 237-260, 1972.

Gomory R.E., Outline of an Algorithm for Integer Solutions to Linear Programs. *Bulletin of the American Mathematics Society*, **64**, 275-278, 1958.

Grossmann I.E., Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques for Process Systems Engineering. submitted to *Journal of Optimization and Engineering*, 2002.

Grossmann I.E., J.P. Caballero and H. Yeomans, Mathematical Programming Approaches to the Synthesis of Chemical Process Systems. *Korean Journal of Chemical Engineering*, **16**(4), 407-426, 1999.

Grossmann, I. E., and Lorenz T. Biegler, "Part II: Future Perspective on Optimization," *Computers and Chemical Engineering*, to appear (2003)

Gupta O.K. and V. Ravindran, Branch and Bound Experiments in Convex Nonlinear Integer Programming. *Management Science*, **31**(12), 1533-1546, 1985.

Heemels, W. P., B. DeSchutter, and A. Bemporad., "On the equivalence of hybrid dynamical models." In *40th IEEE Conference on Decision and Control*, pp. 364—369 (2001)

Hentenryck P.V., *Constraint Satisfaction in Logic Programming*, MIT Press, Cambridge, MA 1989.

Hiriart-Urruty J. and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, New York, 1993.

Hooker J.N., *Logic-Based Methods for Optimization*, John Wiley & Sons, 1999.

Hooker J.N. and M.A. Osorio, Mixed logical/linear programming. *Discrete Applied Mathematics*, **96-97** (1-3). 395-442, 1999.

Horst R. and H. Tuy, *Global Optimization: Deterministic Approaches*, 3rd ed., Springer-Verlag, Berlin, 1996.

ILOG (1999). ILOG OPL Studio 2.1., User's Manual. *ILOG Inc*.

Jackson, J.R. and I.E. Grossmann, "High Level Optimization Model for the Retrofit Planning of Process Networks," *I&EC Research* 41,3762-3770 (2002).

Johnson E.L., G.L. Nemhauser and M.W.P. Savelsbergh, Progress in Linear Programming Based Branch-and-Bound Algorithms: An Exposition, *INFORMS Journal on Computing*, **12**, 2000.

Kallrath J., Mixed Integer Optimization in the Chemical Process Industry: Experience, Potential and Future. *Trans. I. Chem. E*., **78**, Part A, 809-822, 2000.

Kesavan P. and P.I. Barton, Decomposition algorithms for nonconvex mixed-integer nonlinear programs. *American Institute of Chemical Engineering Symposium Series*, (in press) 1999.

Kesavan P. and P.I. Barton, Generalized branch-and-cut framework for mixed-integer nonlinear optimization problems. *Computers Chem. Engng.*, **24**, 1361-1366, 2000.

Kelley Jr. J.E., The Cutting-Plane Method for Solving Convex Programs. *Journal of SIAM*, **8**, 703, 1960.

Kocis G.R. and I.E. Grossmann, Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Ind. Eng. Chem. Res.* **26**, 1869, 1987.

Kocis G.R. and I.E. Grossmann, A Modelling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets. *Computers and Chem. Engng*., **13**(7), 797-819, 1989.

Laird, C. D., L. T. Biegler, B. van Bloemen Waanders, R. A. Bartlett, "Time Dependent Contaminant Source Determination for Municipal Water Networks Using Large Scale Optimization," submitted for publication (2003)

Land A.H. and A.G. Doig, An Automatic Method for Solving Discrete Programming Problems. *Econometrica*, **28**, 497-520,1960.

Lee, S., J.S. Logsdon, M.J. Foral and I.E. Grossmann, "Superstructure Optimization of the Olefin Separation Process," Proceedings ESCAPE-13 (Eds. A. Kraslawski and I. Turunen), pp. 191-196. (2003).

Lee S. and I.E. Grossmann, New Algorithms for Nonlinear Generalized Disjunctive Programming. *Computers Chem. Engng*., **24**, 2125-2141, 2000.

Lee S. and I.E. Grossmann, A Global Optimization Algorithm for Nonconvex Generalized Disjunctive Programming and Applications to Process Systems. *Computers Chem. Engng*., **25**, 1675-1697, 2001.

Leyffer S., Integrating SQP and branch-and-bound for Mixed Integer Nonlinear Programming. *Computational Optimization and Applications* **18**, 295-309, 2001.

Lovász L. and A. Schrijver, Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal on Optimization*, **12**, 166-190, 1991.

Lucia, A., Xu., J. D'Couto, G. C., "Sparse quadratic programming in chemical process optimization," *Annals of Operations Research*, 42, pp. 55-83, 1993

Lucia, A. and J. Xu, "Chemical process optimization using Newton-like methods," *Comp. Chem. Engr*., 14, p. 119 (1990)

Luo, Z-Q, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, 1996

Maravelias, C.T. and I.E. Grossmann, "A New General Continuous-Time State Task Network Formulation for Short Term, Scheduling of Multipurpose Batch Plants," *I&EC Research*, **42**, 3056-3074(2003).

Maravelias, C.T. and I. E. Grossmann, "A Hybrid MILP/CP Decomposition Approach for the Continuous Time Scheduling of Multipurpose Batch Plants," to appear in *Computers and Chemical Engineering* (2004).

McCormick G.P., Computability of Global Solutions to Factorable Nonconvex Programs: Part I – Convex Underestimating Problems. *Mathematical Programming*, **10**, 147-175, 1976.

Narasimhan S., Jordache C., *Data Reconciliation & Gross Error Detection: An Intelligent Use of Process Data,* Gulf Publishing Company, Houston, TX, 2000

Nemhauser G.L. and L.A. Wolsey, *Integer and Combinatorial Optimization*, Wiley-Interscience, New York, 1988.

Nocedal, J. and S. J. Wright, **Numerical Optimization**, Springer, New York (1999)

Poku, M. B., J. D. Kelly, L. T. Biegler, "Nonlinear Programming Algorithms for Process Optimization with Many Degrees of Freedom," to appear, *I & EC Research* (2004)

Quesada I. and I.E. Grossmann, An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers Chem. Engng.,* **16**(10/11), 937-947, 1992.

Quesada I. and I.E. Grossmann, A Global Optimization Algorithm for Linear Fractional and Bilinear Programs. *Journal of Global Optimization*, **6**(1), 39-76, 1995.

Raghunathan, A. U., J. R. Perez-Correa, E. Agosin and L. T. Biegler, "Parameter Estimation in Metabolic Flux Balance Models for Batch Fermentation – Formulations and Solution Using Differential Variational Inequalities," submitted for publication (2004b)

Raghunathan, A. and L. T. Biegler, "MPEC Formulations and Algorithms in Process Engineering," *Computers and Chemical Engineering*, **27**, pp. 1381-1392 (2003a)

Raghunathan, A. U., J. R. Pérez-Correa and L. T. Biegler, "Data Reconciliation and Parameter Estimation in Flux-Balance Analysis," *Biotechnology and Bioengineering*, **84,** pp. 700-709 (2003b)

Raghunathan, A. U. and Lorenz T. Biegler, "Interior point methods for Mathematical Programs with Complementarity Constraints (MPCCs)," submitted for publication (2003c)

Raghunathan, Arvind, M. Soledad Diaz, Lorenz T. Biegler "An MPEC Formulation for Dynamic Optimization of Distillation Operation," submitted for publication, (2004a)

Raman R. and I.E. Grossmann, Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis. *Computers Chem. Engng.,* **15**(2), 73-84, 1991.

Raman R. and I.E. Grossmann, Modelling and Computational Techniques for Logic Based Integer Programming. *Computers Chem. Engng.*, **18**(7), 563-578, 1994.

Ryoo H.S. and N.V. Sahinidis, Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design. *Computers and Chem. Engng*., **19**(5), 551-566, 1995.

Sainz, J., Pizarro, F., Perez-Correa, J.R. and Agosin, E., "Modeling of Yeast Metabolism and Process Dynamics in Batch Fermentation, Biotechnology and Bioengineering, 81, p. 818 (2003)

Sargent, R. W. H., Ding, M., "A new SQP algorithm for large-scale nonlinear programming," *SIAM J. Opt.,* 11, 3, pp. 716-747 (2000)

Sawaya, N.W. and I.E. Grossmann, "A Cutting Plane Method for Solving LinearGeneralized Disjunctive Programming Problems," submitted for publication (2004).

Scholtes, S., "Convergence Properties of a Regularization Scheme for Mathematical Programs with Complementarity Constraints," *SIAM J. Opt.,* 11, 4, pp. 918-936 (2001)

Serth R. W., Heenan W. A., "Gross Error Detection and Data Reconciliation in Stream Metering-Systems," *AIChE Journal*, 32, 5, 733, 1989

Sherali H.D. and W.P. Adams, A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems. *SIAM Journal on Discrete Mathematics*, **3**(3), 411-430, 1990.

Smith E.M.B. and C.C. Pantelides, Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Computers Chem. Engng.*, **21**(1001), S791-S796, 1997.

Stein, O. J. Oldenburg, and W. Marquadt, "Continuous reformulations of discrete-continuous optimization problems." *Computers and Chemical Engineering* ,(2004) accepted for publication.

Stubbs R. and S. Mehrotra, A Branch-and-Cut Method for 0-1 Mixed Convex Programming. *Mathematical Programming*, **86**(3), 515-532, 1999.

Türkay M. and I.E. Grossmann, Logic-based MINLP Algorithms for the Optimal Synthesis of Process Networks. *Computers Chem. Engng.,* **20**(8), 959-978, 1996.

Vanderbei, R.J. and D. F. Shanno. An interior point algorithm for non-convex nonlinear programming. *Comp. Opt. and Applics*. 13, 231-252 (1999)

van der Schaft, A. J., and J.M. Schumacher, "Complemetarity modeling of hybrid systems," *IEEE Transactions on Automatic Control,* 43(4), 1998.

Van Roy T.J. and L.A. Wolsey, Valid Inequalities for Mixed 0-1 Programs. *Discrete Applied Mathematics*, **14**, 199-213, 1986.

Vecchietti A. and I.E. Grossmann, LOGMIP: A Disjunctive 0-1 Nonlinear Optimizer for Process Systems Models. *Computers Chem. Engng.,* **23**, 555-565, 1999.

Viswanathan J. and I.E. Grossmann, A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization. *Computers Chem. Engng.,* **14**, 769, 1990.

Waechter, Andreas and L. T. Biegler, "Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence", *SIAM J. Opt.,* to appear (2003)

Wallace, M., Novello, S. & Schimpf, J. (1997). ECLiPSe: a Platform for Constraint Logic Programming, *ICL Systems Journal, 12 (1)*, 159-200.

Waltz, R., J. Nocedal, J.L. Morales and D. Orban, "Knitro-Direct: A Hybrid Interior Algorithm for Nonlinear Optimization," submitted for publication in Mathematical Programming, (2003)

Westerlund T. and F. Pettersson, An Extended Cutting Plane Method for Solving Convex MINLP Problems. *Computers Chem. Engng.,* **19**, supl., S131-S136, 1995.

Williams H.P., *Mathematical Building in Mathematical Programming*, John Wiley, Chichester, 1985.

Yuan X, S. Zhang, L. Piboleau and S. Domenech, Une Methode d′optimization Nonlineare en Variables Mixtes pour la Conception de Porcedes. *Rairo Recherche Operationnele*, **22**, 331, 1988.

Zamora J.M. and I.E. Grossmann, A Branch and Bound Algorithm for Problems with Concave Univariate, Bilinear and Linear Fractional Terms. *Journal of Global Optimization*, **14**(3), 217-249, 1999.