Chapter 7

GLOBAL OPTIMIZATION AND MIXED INTEGER NONLINEAR PROGRAMMING

Introduction

Minimize: $z = c^T y + f(x)$ (7-1)

Subject to: $Ay + h(x) = 0$

$By + g(x) \leq 0$

$x \in X = \{x | x \in R^n, x^L \leq x \leq x^U\}$

$y \in Y = \{y | y \in \{0, 1\}^m, Ay \leq a\}$

where **x** is a vector of continuous variables that represent the process variables such as flow rates, temperature, pressures, etc., and **y** is a set of binary variables that can be used to define the topology of the system representing the existence or non-existence of different processing units. The nonlinearities in the economic and process models appear in the terms f(x), g(x) and h(x).

If any of the functions in Equations 7-1 are non-linear, the problem corresponds to a mixed integer non-linear programming problem (MINLP). If all functions are linear, it corresponds to a mixed-integer linear programming problem (MILP). If there are no binary variables (0-1) then the problem reduces to a non-linear programming problem (NLP) or linear programming problem (LP) depending on whether the functions are nonlinear or linear. If there are only binary variables present, then it is an integer programing problem (IP).

Most deterministic solution methods for MINLP apply some form of tree-search. There are two broad classes of methods: single-tree and multi-tree methods. Classical single-tree methods include nonlinear branch-and-bound and branch-and-cut methods, while classical multi-tree methods include outer approximation and Benders decomposition. The most efficient class of methods for convex MINLP are hybrid methods that combine the strengths of both classes of classical techniques.

Deterministic optimization of a MINLP problem is usually accomplished using an algorithm like the branch and bound or the inner-outer method. These algorithms solve a series of NLP problems that typically use the generalized reduced gradient method or successive (sequential) quadratic programming. These NLP algorithms have a super-rate of convergence and locate the optimum in 2n steps for quadratic functions.

Depending on the character of the objective function and constraints, the NLP algorithms will locate a point better than the starting point. If the objective function and constraint equations are second order differentiable, the algorithms will locate extreme points (maxima, minima or saddle points. Necessary conditions are used to determine extreme points, and sufficient condition are used to determine the character of the extreme points. If the objective function and constraints are convex functions the extreme point located is a global optimum.

To locate the extreme points of a nonlinear programming problem, the Lagrange function is used. To form this function the constraint equations are multiplied by Lagrange multipliers and added to the objective function. The inequality constraint equations have been converted equality constraint equations by incorporating slack variables. This unconstrained equation, called the Lagrange function, is partially differentiated with respect to the independent variables and the Lagrange multipliers, and the resulting set of equations are set = 0. Differentiating the Lagrange function with respect to the Lagrange multipliers returns the constraint equations. Solutions to the set of equations are extreme points for the constrained problem, and they are called Kuhn-Tucker points. Extreme points can be maximum, minimum or saddle points that are located by this necessary condition. Sufficient conditions are required to determine the character of the extreme points. See the details of this development in Chapter 2.

Examples of illustrative MINLP problems are given by Belotti, et.al., 2012 for the design of multiproduct batch plants and design of water distribution network, and they illustrate problem reformulation, convex relaxation, relaxation of structured nonconvex sets, and heuristics. Byrne and Bogle 2000 have examples for optimization of an interval process flow sheet and the classic Haverly pooling problem. Examples of heat exchange and reactor networks, blending and pooling, and several for chemical process are given by Sahinidis, 2005. Trespalacios and Grossmann 2014 have an example for a process superstructure optimization.

Nonconvex MINLPs pose additional challenges, because they contain nonconvex functions in the objective or the constraints. Even when the integer variables are relaxed to be continuous, the feasible region is generally nonconvex, resulting in many local minima. A range of approaches are used to tackle this challenging class of problems, they include piecewise linear approximations, generic strategies for obtaining convex relaxations of nonconvex functions, spatial branch-and-bound methods, and a small sample of techniques that exploit types of nonconvex structures to obtain improved convex relaxations, Belotti, et.al., 2012. Several strategies for solving nonconvex MINLPs are reported by Trespalacios and Grossmann 2014 including relaxation and several types of bound tightening.

Equation (7-1) is said to be a NP-hard combinatorial problem, because it includes MILP and its solution typically requires searching enormous search trees. For (7-1) to be decidable, either X is compact or that the problem functions are convex. Nonconvex integer optimization problems are in general undecidable, Belotti, et.al., 2012. Jeroslow1973 describes a study of a class of integer programming problems with square of variables in constraints that "no computing device can be programmed to compute the optimum criteria value for all problems in this class." Jeroslow, 1974 reports on trivial integer programs unsolvable by branch and bound.

In computational complexity theory, NP (for nondeterministic polynomial time) is a complexity class that is used to describe certain types of decision problems. A formal definition of NP is the set of decision problems solvable in polynomial time by a theoretical non-deterministic

Turing machine. In theoretical computer science, a Turing machine is a theoretical machine that is used in thought experiments to examine the abilities and limitations of computers. A decision problem is solved using an algorithm. For NP, polynomial time refers to the increasing number of machine operations needed by an algorithm relative to the size of the problem. Decision problems are commonly categorized into complexity classes (such as NP) based on the fastest known machine algorithms. An example of an NP-hard problem is the optimization problem of finding the least-cost cyclic route through all nodes of a weighted graph, the traveling salesman problem. (Wikipedia, NP-hardness, accessed 4-19-18).

Global Optimization Algorithms

Global optimization is the task of finding the absolutely best set of values of variables to optimize an objective function (Gray et al., 1997). Global optimization problems are typically difficult to solve. Global optimization problems are solved by extension of ideas from local optimization. These algorithms are integrated into computer programs for solving MINLP problems. Both Pinter, 2014 and Trespalacios and Grossmann 2014 provide reviews of the more successful global algorithms and results of robustness *vs.* efficiency in practically motivated test problems.

Global optimization is a branch of applied mathematics and numerical analysis that deals with the global optimization of a function or a set of functions according to some criteria. Typically, a set of bound and more general constraints is also present, and the decision variables are optimized considering these constraints. Global optimization is distinguished from regular optimization by its focus on finding the maximum or minimum over all input values, as opposed to finding local minima or maxima. The *Journal of Global Optimization*, Springer, is one source of numerous publications on the multiplicity of methods tried to solve global optimization problems.

Significant research has been spent developing algorithms that find the global optimum of a problem directly. This would eliminate using the procedure of finding all the local optima and then comparing these local optima to find the largest one, the "global optimum".

Global optimization algorithms are either deterministic or stochastic methods. The most successful deterministic strategies include inner and outer approximation methods, branch and bound methods, cutting plane methods and interval bounding methods. Successful stochastic strategies include random search, genetic algorithms and simulated annealing.

Deterministic Methods: Global optimization uses several optimization algorithms together to locate the global optimum of a mixed integer nonlinear programming problem directly. The Branch and Bound algorithm can be used to separate the original problem into sub-problems that can be eliminated by showing these sub-problems that cannot lead to better points. The Bound Constraint Approximation algorithm rewrites the constraints in a linear approximate form, so a MILP solver can be used to give an approximate solution to the original problem. Penalty and barrier functions can be used for constraints that cannot be linearized. Branching is performed on local optima to proceed to the global optimum using a sequence of feasible sets (boxes). Another algorithm, Box Reduction uses constraint propagation, interval analysis, convex relations and duality arguments involving Lagrange multipliers. The Interval Analysis algorithm attempts to reduce the interval on the independent variables that contains the global optimum. The Leading Global Optimization Solver BARON (Branch and Reduce Optimization Navigator) developed by Professor Nikolaos V. Sahinidis and colleagues at the University of Illinois is a GAMS solver. Global optimization solvers are currently in the code-testing phase of development that occurred 20 years ago for NLP solvers.

Stochastic Methods: The more successful stochastic strategies include random search, genetic algorithms and simulated annealing. Random search is a stochastic method that places measurements (evaluation of the objective function) randomly in the initial intervals of the independent variables. Depending on the number of experiments used, the values of the objective function are ranked, and it can be said statistically that the maximum (or minimum) is in the top x percent with a y probability. The values of the initial intervals can be adjusted based on these results to have smaller region to search, and random measurements are placed in the new region (creeping random search). See Pike, 2013.

Genetic algorithms, annealing algorithms, tabu search, artificial neural networks, among others, use randomized search techniques for finding near optimal solutions of combinatorial optimization problems (Pardalos and Resende, 2002 and Schaffer, 2012). The idea behind using artificial neural networks is to map the optimization problem into a highly-interconnected network of neurons, and a particular configuration of neurons being on or off determines the value of the objective function. The procedure uses an activation function to transform the neurons to locate the configuration that approaches the global solution of the objective function. A sigmoid function is said to be the most used activation function in the artificial neural network literature (Trafalis and Kaspa 2002.)

Simulated annealing is a family of randomized algorithms for locating near optimal solutions of combinatorial optimization problems using the idea of annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Slow cooling is used as an analogy to decrease in the probability of accepting worse solutions as it explores the solution space because it allows for a more extensive search for the optimal solution. Steps with improvements are accepted and ones that do not improve the value of the objective function are accepted within a certain probability. The goal is to bring the system, from an arbitrary initial state to a state with the minimum possible thermodynamic free energy. Threshold algorithms are used to move to improved values of the objective function and are described by Aarts and Ten Eikelder 2002.

Genetic algorithms (Goldberg, 1989) use search heuristic that mimics the process of natural selection to generate useful solutions to optimization problems. The initial solution starts from a population of randomly generated individuals and moves based on heuristics. New solutions are combined with old solutions to generate improved solutions, ones that move to the optimum of the objective function. The algorithm terminates when a maximum number of iterations is reached, or a satisfactory value of the objective function has been obtained.

A comparison of deterministic and stochastic approaches for global optimization for chemical process design by Choi and Manousiouthakis, 2002 describes pseudocode for a simulated annealing and a genetic algorithm among other deterministic and stochastic ones. Their simulated annealing algorithm is reproduced below to illustrate stochastic methods.

Consider a collection of atoms in equilibrium at a given temperature, T. Displacement of an atom causes a change ΔE in the energy of the system. If ΔE<0, the displacement is accepted. If ΔE >0, the probability that the displacement is accepted is exp (-ΔE/kT) where k is the Boltzmann constant. The process can be simulated for optimization as follows.

For minimization of objective function f(**x**)
1. Take $\mathbf{x}^{new}$ randomly.
2. If =Δf = f($\mathbf{x}^{new}$)– f($\mathbf{x}^{old}$) < 0 accept $\mathbf{x}^{new}$.
Otherwise,
a. Take a random number w∈ [0, 1].
b. If w ≤ exp (-ΔE/kT), then accept $\mathbf{x}^{new}$
Otherwise, $\mathbf{x}^{old} = \mathbf{x}^{new}$.
Control T, and repeat.

They conclude that chemical process optimization problems are high rank, non-complex problems, and the guarantee of global optimally is still computationally too expensive. Stochastic algorithms inevitably take forever to obtain a solution where optimality is guaranteed.

Interval Methods: These methods start by bounding the intervals on the independent variables that contain the global optimum. Then they proceed to reduce the bounds on these variables by various means to have final intervals of the desired precision containing the global optimum. These types of methods evaluate each constraint with the current variable bounds and try to improve bounds by maintaining feasibility in the constraints. A recent method uses pairs of constraints instead of individual constraints to infer bounds. Different techniques have been developed to infer bounds on MILP problems and on MINLP problems. Details are provided by Trespalacios and Grossmann 2014.

Global Optimization for Chemical Process Systems

Deterministic optimization of a MINLP problem for a chemical process system is usually accomplished using an algorithm like the branch and bound or the inner-outer method. These algorithms solve a series of NLP problems that typically use the generalized reduced gradient method (GRG) or successive (sequential) quadratic programming (SQP). These NLP algorithms have a super-rate of convergence and locate the optimum in 2n steps for quadratic functions.

Depending on the character of the objective function and constraints, the NLP algorithms will locate a point better than the starting point. If the objective function and constraint equations are second order differentiable, the algorithms will locate extreme points (maxima, minima or saddle points). Necessary conditions are used to determine extreme points, and sufficient condition are used to determine the character of the extreme points. If the objective function is concave and the constraint equations are convex the extreme point is a minimum.

Branch and Bound Methods: These methods use a systematic enumeration of candidate solutions that are thought of as forming a tree with the full set of solutions at the top of the tree. The algorithm explores branches of this tree that represent subsets of the solution set. Each branch is checked against upper and lower estimated bounds on the optimal solution and branches are discarded if they cannot produce a better solution than the best one found so far by the algorithm. Nonlinear branch and bound is an extension to the well-known linear branch and bound algorithm. To find optimality, the method performs a tree search on the integer variables. It first solves the continuous relaxation problem (r-MINLP). If the solution yields integer values to all integer variables, then it is optimal, and the algorithm stops. If it is not, a branching heuristic is used to select an integer variable whose value at the current node is not integer ($y_i = y_i^0$). A branching is performed in this variable, giving rise to two new NLP problems. One NLP includes the bound $y_i \leq y_i^0$ while the other one $y_i \geq y_i^0$ i.e., $y_i = 0$ or $y_i = 1$ if the integer variables are binary (0 – 1) variables.

This procedure is repeated until the tree search is exhausted. If an integer feasible solution is found, i.e., the solution provides integer values to all the integer variables, then it provides an upper bound. There are two cases in which some of the nodes are pruned, which make the branch and bound method faster than enumerating every node. The first case in which a node is pruned occurs when the NLP corresponding to the node is infeasible. The second case occurs when the solution of the NLP of the node is larger than the current upper bound for minimization. A detailed description of this algorithm is given by Trespalacios and Grossmann, 2014 and Schaffer, 2012.

The general form of a convex MINLP model is:

$$\min z = f(x, y) \tag{7-2}$$

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

$$x \in X$$

$$y \in Y$$

where f and g are twice continuously differentiable functions and are convex functions, x are the continuous variables, and y the discrete variables. The Kuhn-Tucker conditions are necessary and sufficient for a global (absolute) maximum (Cooper, 1981). Note: Theorem 20, If f(x) is strictly a concave function and $g_i(x)$ are convex functions, for the NLP, (max f(x) subject to $g_i(x) \leq b_i$, i = 1, 2, ..., m,) which are continuous and differentiable, the Kuhn-Tucker conditions are sufficient as well as well as necessary for an absolute maximum, ref. (Cooper, 1981)

Branch and Bound Algorithm. Nonlinear branch and bound is based on the branch and bound algorithm for MILP. The idea of the branch-and-bound technique is to divide and conquer. If the original problem is very large, then it would be difficult to solve it directly; and hence it is divided into smaller and smaller subproblems (nodes) until these subproblems can be solved easily or conquered.

The solution to four NLP problems is used in the branch and bound method and other methods. One is a linear approximation to the convex MINLP about point P (p=1, 2, ...P) and is called a relaxation of the MINLP (M-MIP). The second is the continuous relaxation of the MINLP and is called (r-MINLP) where the integer variables are treated as continuous and gives the lower bound to the MINLP. The third is for a fixed y^p in the convex MINLP; this NLP (fx-MINLP) and is any feasible solutions to this NLP (fx-MINLP) is an upper bound on the MINLP. The fourth is when there is not a feasible solution to the NLP (fx-MINLP), the following feasible NLP (feas-MINLP) is solved to minimize the infeasibility of the most violated constraint, $g(x, y) \le eu$ where e is a vector of ones (Trespalacios and Grossmann 2014).

• A linear approximation to the convex MINLP about point P (p=1, 2, ...P) is called a relaxation of the MINLP (M-MIP) and is given by:

min a

$$\text{s.t.} \quad f(x^p,y^p) + \nabla f(x^p,y^p)\bullet[\,(x - x^p), (y - y^p)] \le a \qquad \text{for } p = 1,2, \dots P \qquad \text{M-MIP}$$

$$g(x^p,y^p) + \nabla g(x^p,y^p)\bullet[\,(x - x^p), (y - y^p)] \le 0$$

The linear approximation provides a lower bound to the MINLP because of the convexity of the MINLP. If this relaxation is infeasible, then MINLP is also infeasible. If the solution of the relaxation is integer, then it also solves the MINLP.

• The continuous relaxation of the MINLP is given by the following NLP (r-MINLP):

$$\min z = f(x, y)$$

$$\text{s.t.} \quad g(x, y) \le 0 \qquad \text{r-MINLP}$$

$$x \in X$$

$$y \in Y_R$$

where the integer variables are treated as continuous. Y_R is a continuous relaxation of Y with upper and lower bounds, $y^{lo} \le y \le y^{up}$. Any feasible solutions to this NLP (r-MINLP) is a lower bound on the MINLP.

• For a fixed y^p in the convex MINLP, the NLP (fx-MINLP) is:

$$\min z = f(x, y^p)$$

$$\text{s.t.} \quad g(x, y^p) \le 0 \qquad x \in X \qquad \text{fx-MINLP}$$

Any feasible solutions to this NLP (fx-MINLP) is an upper bound on the MINLP.

• When there is not a feasible solution to the NLP (fx-MINLP), the following feasible NLP (feas-MINLP) is solved to minimize the infeasibility of the most violated constraint.

min u

s.t. $g(x, y) \leq eu$ feas-MINLP

$x \in X$

$u \in R$

where e is a vector of ones.

For a node N_p, let z^p denote the optimal value of the corresponding NLP_p, and (x^p, y^p) its solution. Let L be the set of nodes to be solved, and NLP_0 be (r-MINLP), continuous relaxation of the MINLP. Let z^{lo} and z^{up} be, respectively, a lower and upper bound of the optimal value of the objective function z^*. A tolerance for termination $\varepsilon > 0$ is specified.

For Node Selection to start and continue an algorithm, select a noninteger basis variable y_i in the MINLP problem (initially, the r-MINLP relaxation solution). Construct NLP_p^1 and NLP_p^2 by adding one of the constraints: $y_i \leq y_i^p$ and $y_i \geq y_i^p$ in each of the problems (or $y_i = 0$ and $y_i = 1$, if y_i is a binary variable).

If there are more than one noninteger basis variables in the problem, then any one of them can be selected for branching. The solution may move more rapidly by selecting the variable with the largest fractional value.

Solving NLP_p^1 and NLP_p^2 begin the formation of a tree structure. If an integer feasible solution is found, i.e., the solution provides integer values to all the integer variables, then it provides an upper bound. There are two cases in which some of the nodes are pruned i.e., no further branching, which make the branch and bound method faster than enumerating every node. The first case in which a node is pruned occurs when the NLP corresponding to the node is infeasible. The second case occurs when the solution of the NLP of the node is larger than the current upper bound for minimization (Trespalacios and Grossmann 2014). Continuing, this procedure is repeated until the tree search is exhausted.

Begin the branch and bound Algorithm for minimizing. The algorithm uses a series of steps as follows.

- Step 0: Initialization: Solve the continuous relaxation NLP, (r-MINLP).

$L = N_0$, $z^{up} = \infty$, $(x^*, y^*) = 0$

- Step 1: Terminate?

If the continuous relaxation solution NLP_0 (r-MINLP), yields integer values to all integer variables ($L = 0$), then (x^*, y^*) is optimal and the algorithm stops.

If an integer solution has not been found, then the r-MINLP solution, NLP_0, provides an upper bound, z^{up}, to the MINLP because the optimal integer solution cannot have an objective function value larger than the r-MINLP solution. The imposition of integer restriction on $\boldsymbol{y}$ can only decrease the optimal value of the MINLP.

A lower bound z^{lo} for the optimal objective function value is equal to the objective value at some point that is feasible for the MINLP problem. This could be where all the variables are zero or some comparable solution that satisfies all the constraints and that will surely be smaller than the final optimal value.

If no such feasible point is readily known for the lower bound, set $z^{lo} = -\infty$. This lower bound solution is designated as the incumbent solution. This means that it is the best MINLP solution obtained so far. When a better integer feasible point is obtained as the solution proceeds, then that would be the new incumbent solution.

The linear approximation provides a lower bound to the MINLP (M-MIP) because of the convexity of the MINLP. If this relaxation is infeasible, then MINLP is also infeasible. If the solution of the relaxation is integer, then it also solves the MINLP.

• Step 2: Node Selection

Select a noninteger basis variable y_i in the MINLP problem (initially, the r-MINLP relaxation solution). A branching is performed in this variable, giving rise to two new NLP problems. Construct NLP_p^1 and NLP_p^2 by adding one of the constraints $y_i \leq yip$ and $y_i > yip$ in each of the problems (or $y_i = 0$ and $y_i = 1$, if y_i is a binary variable).

One NLP includes the bound $y_i < yi^0$ while the other one $y_i > y_i^0$ i.e., yi = 0 or $y_i = 1$ if the integer variables are binary (0 – 1) variables.

If there is more than one noninteger basis variables in the problem, then any one of them can be selected for branching. The solution may move more rapidly by selecting the variable with the largest fractional value.

• Step 3: Branch. (Partition problem into two subsets)

Branching is accomplished by adding constraints to the MINLP problem to exclude the noninteger values of the chosen basis variable. For example, if the current solution has the values $\boldsymbol{y} = [0, 2.5, 3]$, then this set is partitioned further into two subsets by adding an additional constraint to exclude the noninteger value of the variable. (In this case, y_2). The additional constraints for the two subsets would be $y_2 \leq 2$ and $y_2 \geq 3$ respectively.

• Step 4: Bound. (Solve NLP's, NLP_p^1 and NLP_p^2, from subsets)

Solve the two new problems, NLP_p^1 and NLP_p^2 that are obtained by appending the extra constraint as a result of Step 3. These are designated as subsets, and their resulting optimal values (if they are not infeasible) would be the upper bound z^{up} for that branch when the subset is developed. Additional integer constraints are added in expanding branches

Step 5: Fathom (Prune)

Tests for the solution of NLP_p^1 and NLP_p^2 to determine if further branching is required.

(a) $z^{up} \leq z^{lo}$, i.e. NLP objective function value is less than the lower bound, and no further evaluations are needed.

(b) The NLP has no feasible points, and no further evaluations are needed.

(c) If z^{up} is an integer feasible solution and $z^{up} > z^{lo}$, then this is the new incumbent solution, since it is the best integer solution obtained thus far.

Select a subset among those from Step 4 that has noninteger values for branching. If all subsets have been fathomed or pruned, the incumbent solution is optimal for MINLP. Otherwise, return to Step 2.

Example 7-1. This is an example of branch and bound for a MINLP problem modified from Sahinidis, N., 2005. The diagram in Figure 7-1 shows the constraint equations and the objective function with P as a parameter.

max: $P = +x_1 + x_2$

s.t. $x_1 x_2 \leq 4$

$0 \leq x_1 \leq 6$

$0 \leq x_2 \leq 4$

Continuous relaxation solution:

$x_1 = 6$, $x_2 = 0.67$, $P = 6.67$, upper bound = 6.67, lower bound = 0.0

Branching on x_2 using the two constraints added the original MINLP:

$x_2 \geq 1$, $x_2 \leq 1$

max: $P = + x_1 + x_2$	max: $P = +x_1 + x_2$
s.t. $x_1 x_2 \leq 4$	s.t. $x_1 x_2 \leq 4$
$0 \leq x_1 \leq 6$	$0 \leq x_1 \leq 6$
$0 \leq x_2 \leq 4$	$0 \leq x_2 \leq 4$
$x_2 \geq 1$	$x_2 \leq 1$

The solutions to the above two problems are:

$x_1 = 5$, $x_2 = 1$, $P = 6$, upper bound = 6. lower bound $x_1 = 6$, for problem with $x_2 \geq 1$

$x_1 = 6$, $x_2 = 0.67$, $P = 6.67$, upper bound = 6.67, lower bound = 6 for problem with $x_2 \leq 1$.

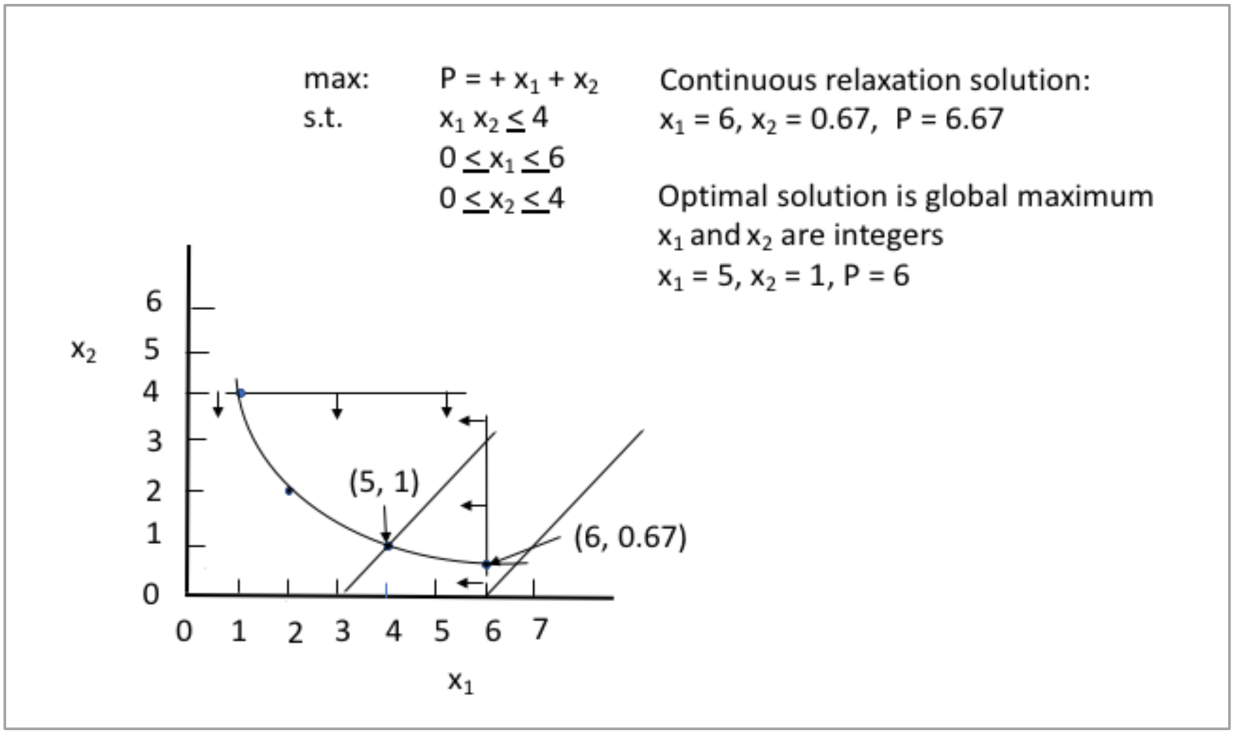

Figure 7-1 Diagram of Example Problem 7-1

The global maximum is the optimal solution of MINLP problem with constraint of $x_2 \geq 1$. Both x_1 and x_2 are integers. This simple problem only had one noninteger variable for branching, x_2, since x_1 was a integer from the continuous relaxation solution. For more complex MINLPs the procedure for branching and bounding is the same, select a noninteger variable and form two new MINLP's with inequality constraints.

Selection of branching variable is a crucial component of branch-and-bound. A simple branching rule is to select the variable with the largest integer violation for branching which is known as maximum fractional branching. In practice however, this branching rule is not efficient: it performs about as well as randomly selecting a branching variable. Details on five efficient methods are described by Belotti et al, 2012. The more successful branching rules estimate the change in the lower bound after branching including pseudo-costs branching, reliability branching and branching on general disjunctions.

Node selection strategies refers to important decisions about which node should be solved next. The goal of this strategy is to find a good feasible solution quickly in order to reduce the upper bound, and to prove optimality of the current incumbent $x*$ by increasing the lower bound as quickly as possible. Two popular strategies, depth-first search and best-bound search, have strengths and weaknesses as described by Belotti et al, 2012. Also, they present two hybrid schemes that aim to overcome the weaknesses of these two strategies are described, best bound search and hybrid search.

Other methods for solving MINLP problems include cutting planes, multi-tree methods, outer approximation, generalized Benders decomposition and single-tree methods Disjunctive cuts are used in the class of convex MINLPs where the only nonconvex constraints are represented by integer variables, and these non-convexities are resolved by integer branching, which represents a specific class of disjunctions. Generic relaxation strategies are methods for finding a relaxation to exploit the structure of the problem. For a broad class of MINLP problems, the objective function and the constraints are nonlinear but factorable, in other words, they can be expressed as the sum of products of unary functions. See details given by Belotti et al, 2012 and Grossmann and Trespalacios, 2013.

Inner and Outer Approximation Methods: Outer-approximation (OA) makes use of two problems: (M-MIP) and (fx-MINLP). The approach is to use the approximate linear problem (M-MIP) to find a lower bound (z_{lo}) and obtain an integer solution to the approximate problem (y_p). This lower bounding problem is called master problem. For the subproblem, the binary variables y_p are fixed, and then (fx-MINLP) is solved. If the solution to (fx-MINLP) is feasible, then it provides an upper bound. If it is not, (feas-MINLP) is solved to provide information about the subproblem, and an inequality that cuts off that integer solution is added. This method is performed iteratively until the gap of z_{lo} and z_{up} (the best upper bound) is less than the specified tolerance. At each iteration, the sub-problem (either (fx-MINLP) or (feas-MINLP)) provides a solution (x_p, y_p) that is included in the master problem (M-MIP) to improve the approximation. Since the function linearizations are accumulated, the lower bounding problem (or master problem) yields a nondecreasing lower bound ($z_{lo,1} \leq z_{lo,2} \leq \ldots \leq z_{lop}$). The outer-approximation algorithm is described in more detail by Trespalacios and Grossmann 2014.

Generalized Benders Decomposition (GBD): This method is similar to the OA method, but they differ in the linear master problem. The master problem of the GBD considers the discrete variables $y \in Y$, and the active inequalities $J_p = \{ j | g_j(x_p, y_p) = 0\}$. Details for this algorithm are given by Trespalacios and Grossmann 2014.

Extended Cutting Plane (ECP): This method is similar to the OA method, but it avoids solving NLP sub-problems. At a given solution of the master MILP (M-MIP), all the constraints are linearized. A subset of the most violated linearized constraints is then added to the master problem. Convergence is achieved when the maximum violation lies within a specified tolerance. The algorithm has nondecreasing a lower bound after each iteration. The main strength of the method is that it relies solely in the solution of MILPs. Similarly, to the OA method, it solves the problem in one iteration if $f(x, y)$, and $g(x, y)$ are linear. Two downsides in the algorithm are that convergence can be slow and that the algorithm does not provide an upper bound (or feasible solution) until it converges (Trespalacios and Grossmann 2014).

Heuristic Search Techniques: Heuristics have been developed for solving MINLPs when applications are too large to be solved. Very large problems generate a huge search tree or must be solved in real time. In these situations, it is more desirable to obtain a good solution quickly than to wait for an optimal solution. It is necessary to resort to heuristic search techniques that provide a feasible point without any optimality guarantees. Heuristics can accelerate rigoristic techniques by quickly identifying an incumbent with a low value of the objective function. This upper bound can then be used to prune a larger number of the nodes in the branch- and-bound algorithm. Two classes of heuristic search techniques are probabilistic search and deterministic search. Probabilistic search refers to techniques that require at each iteration a random choice of a candidate solution or parameters that determine a solution. Deterministic techniques, for example can run branch-and-bound for a fixed time or fixed number of nodes or until it finds its first incumbent. Heuristics can be of two types: search heuristics, which search for a solution without the help of any known solutions, and improvement heuristics, which improve upon a given solution or a set of solutions. Details are given by Belotti et al, 2012.

An important area of application of mathematical programming is optimization in the synthesis of process flow sheets. A general overview of the MINLP approach and algorithms for process synthesis was presented by Grossmann, (1990). A basic understanding of several algorithmic techniques as well as the relative strengths, weaknesses and difficulties have been detailed. Also, it was shown that effective modelling schemes and solution strategies can play a crucial role in the successful application of techniques. According to the author, the major steps involved in the MINLP approach include postulating a superstructure that has several feasible and optimal design alternatives. This superstructure is then modelled as an MINLP problem in which 0-1 variables are assigned to the potential existence of units, and continuous variables to the flows, temp, pressure, sizes, etc. Then the optimal design is extracted from the superstructure by solving the MINLP problem. Good MINLP formulation can be done by keeping the problem as linear and convex as possible, and by having a tight NLP relaxation. In order to increase the reliability and efficiency of the solution procedure, it is also important to recognize the special structure and properties that characterize the optimal synthesis of process systems.

Non-Convex MINLPs

A common approach for approximately solving MINLPs with nonconvex functions is to replace the nonlinear functions with piecewise linear approximations, leading to an approximation that can be solved by mixed-integer linear programming solvers. However, A very large literature on global optimization includes several textbooks and review articles. MINLP is one of the most complex and active fields in optimization according to Trespalacios and Grossmann 2014. Accurate modeling of many industrial problems, particularly in chemical engineering, requires the use of nonconvex constraints. They described spatial branch and bound as the most widely used method to solve non- convex MINLP. Two main concepts used in most applications are relaxations of factorable formulations and bounds tightening. Their description includes feasibility-based, optimality-based, reduced-cost, and probing bound tightening methods.

Spatial branch-and-bound is the best-known method for solving nonconvex MINLP problems, according to Belotti, et.al., 2012. Most modern MINLP solvers designed for nonconvex problems utilize a combination of the techniques, in particular, they are branch-and-bound algorithms with at least one rudimentary bound-tightening technique and a lower-bounding procedure. Methods used by several established MINLP solvers are described, including BARON, COCONUT, COUENNE and LindoGlobal.

In relaxations of structured nonconvex constraints, this approach is used to relax any constraint containing a nonlinear function that can be factored into simpler primitive functions which have known relaxations. Then this relaxation is refined after spatial branching. When combined with relaxation and branching on integer variables, this leads to algorithms that can (theoretically) solve almost any MINLP with explicitly given nonlinear constraints. The drawback of this general approach is that the relaxation obtained may be weak compared with the tightest possible relaxation, and the convex hull of feasible solutions leads to an impractically large branch-and-bound search tree, Belotti, et.al., 2012.

GAMS (General Algebraic Modeling System) Programming Language

The General Algebraic Modeling System (GAMS) is a high-level modeling language for mathematical programming and optimization. It consists of a language compiler and integrated high-performance solvers. GAMS is tailored for complex, large scale modeling applications, and allows building of large maintainable models that can be adapted quickly to new situations. The GAMS offer a wide range of solvers that allow the optimization based on type of problem. These include LP, NLP, MILP, MINLP and Global optimization solvers. The GAMS (General Algebraic Modeling System) programming language was developed by the GAMS Development Corporation 1217 Potomac Street, NW, Washington, D.C. 20007 (http://www.gams.com).

GAMS is specifically designed for solving linear, nonlinear and mixed integer optimization problems. The system is especially useful with large, complex problems. GAMS is available for use on personal computers, workstations, mainframes and supercomputers. GAMS is able to formulate models in many different types of problem classes and switching from one model type to another can be done with a minimum of effort. The same data, variables, and equations can be used in different types of models at the same time.

GAMS model types include Linear Programming (LP), Mixed-Integer Programming (MIP), Mixed-Integer Non-Linear Programming (MINLP), and different forms of Non-Linear Programming (NLP). There are over 30 solvers (optimization codes) that can be selected to solve these programming problems. Note, “programming,” means “scheduling” and not “computer programming.” An extensive list of solvers can be found at GAMS website (www.GAMS.com) for solving LP, NLP, MIP, MILP and MINLP problems. The solvers used to solve the global optimization problem in the Chemical Complex Analysis System were BARON and LINDOGLOBA.

GAMS Distribution 25.1.1is currently available (5-19-18) for download from the GAMS web site www.GAMS.com without charge. GAMS will operate as a free demo system without a valid GAMS license. The model limits in demo mode are 300 constraints and variables, 2000 nonzero elements, (of which 1000 can be nonlinear), 50 discrete variables (including semi continuous, semi integer and member of SOS-Sets) with additional global solver limits of 10 constraints and variables. There are the installation notes for Windows, Mac, and UNIX. The GAMS distribution includes the GAMS Manuals in electronic form, and hard copies can be ordered through Amazon.

The NEOS Server for Optimization hosted by the Argonne National Laboratory is an open and free to use server for solving optimization problems (NEOS, 2010). The optimization solvers at NEOS represent the state-of-the-art in optimization software. Optimization problems are solved automatically with minimal input from the user. The users only need a definition of the optimization problem, and all additional information required by the optimization solver is determined automatically by the server. For example, the solver choice for MINLP is required, but the sub-choices for LP and NLP need not be specified in the server.

MINLP Solver Performance

An overview of the start-of-the-art in software for the solution of mixed integer nonlinear programs (MINLP) is given by Bussieck and Vigerske, 2014, of GAMS that describes various features of embedded and independent solvers with a concise description for each solver to provide to guide the selection of a best solver for a particular MINLP problem. They establish several groupings with respect to various features and give concise individual descriptions for each solver. The objective is to provided information to guide the selection of a best solver for a particular MINLP problem. Global optimization of MINLP requires an effective algorithm or combination of algorithms, usually LP, MIP and NLP, implemented in programming languages, and run on a computer with an operating system for linear or parallel operations. Over time there have been research results reported on efficient algorithms for sets of problems. The sets of problems have become comparable in size to industrial plants, and algorithms (solvers) have improved correspondingly. Algorithms for solving MINLPs are built by combining algorithms from linear programming, integer programming, and nonlinear programming, e.g., branch and bound, outer approximation, local search, global optimization. MINLP solvers often combine LP, MIP, and NLP solvers. Some solvers that guarantee global optimal solutions for general convex MINLPs but not for general nonconvex MINLP. In case of a nonconvex MINLP, these solvers can still be used as a heuristic. Especially branch and bound based algorithms that use NLPs for bounding often find good solutions. Solvers that also guarantee global optimality for nonconvex general MINLPs require an algebraic representation of the functions f(x, y) and g(x, y) for the computation of convex envelopes and underestimators. Each function needs to be provided as a composition of basic arithmetic operations and functions (addition, multiplication, power, exponential, trigonometric, ...) on constants and variables. Over time there have been research results reported on efficient algorithms for sets of problems. The sets of problems have become comparable in size to industrial plants, and algorithms (solvers) have improved correspondingly.

A review of deterministic software for solving convex MINLP problems was given by Kronqvista et al., 2018. It included a comprehensive comparison of a large selection of commonly available solvers. MINLPLib included a test set of 366 convex MINLP instances. All MINLP instances were classified as convex in the problem library. A summary of the most common methods for solving convex MINLP problems was given to better highlight the differences between the solvers. To show how the solvers perform on problems with different properties, the test set was divided into subsets based on the integer relaxation gap, degree of nonlinearity, and the relative number of discrete variables. The results presented provide guidelines on how well suited a specific solver or method is for particular types of MINLP problems. BARON was said to be very efficient at identifying problems as convex since it is able to deal with these problems in such an efficient manner. The solvers were used programming languages GAMS, AMPL, and AIMMS.

Comparisons of global optimization programs (solvers) are given for a chemical production complex optimization with new processes for chemicals from biomass (Sengupta and Pike, 2012.) The optimal structure was determined from the superstructure of global optimization problem in the Chemical Complex Analysis System using five different solvers from the NEOS server. These were DICOPT, SBB, BARON, ALPHAECP and LINDOGLOBAL. Two of these solvers were listed exclusively under global solvers that accepted GAMS input (BARON and LINDOGLOBAL), and the other three were listed under MINLP solvers (DICOPT, SBB, ALPHAECP). The results for computation time and solver status from the NEOS server solution are given in Table 10-1 from Sengupta and Pike, 2012. The SBB, DICOPT and BARON gave a normal completion with identical solutions for the objective value. Computational, generation and execution times were comparable. The LINDOGLOBAL was unable to solve because of an iteration interrupt. The ALPHAECP gave a normal completion with infeasible solution. Table 10-2 gives the comparison of the solution using SBB in the NEOS server and the local machine, an Intel PC, and the results were the same.

The most common method to solve nonconvex MINLPs to ε-global optimality is spatial branch-and-bound that recursively divides the original problem into subproblems on smaller domains until the individual subproblems are easy to solve (Vigerske and Gleixnerscip, 2016). Bounding is used to decide early whether improving solutions can be found in a subtree. These bounds are computed from a convex relaxation of the problem, that is obtained by dropping the integrality requirements and relaxing nonlinear constraints by a convex or even polyhedral outer approximation. Branching, i.e., the division into subproblems, is typically performed on discrete variables that take a fractional value in the relaxation solution and on variables that are involved in nonconvex terms of violated nonlinear constraints. The restricted domains allow for tighter relaxations in the generated subproblems.

Over the last decades, substantial progress has been made in the solvability of both mixed-integer linear programs and nonconvex nonlinear programs. The integration of MIP and global optimization of NLPs and the development of new algorithms unique to MINLP have led to a variety of general-purpose software packages for the solution of medium-size (nonconvex) MINLPs. One of the first of this kind and still actively maintained and improved is BARON which implements a branch-and-bound algorithm employing LP relaxations. Later, Lindo, Inc., added global solving capabilities to their Lindo API solver suite. An open-source implementation of a global optimization solver is available with Couenne (neos-server.org/neos/solvers). A branch-and-bound algorithm based on a mixed-integer linear relaxation is implemented in the solver ANTIGONE (Vigerske and Gleixnerscip, 2016).

Table 7-1 Comparison of Solvers in NEOS Server for Optimal Solution (Sengupta and Pike, 2012)

Solver	**SBB (MINLP)**	**DICOPT (MINLP)**	**ALPHAECP (MINLP)**	**BARON (Global)**	**LINDOGLOBAL (Global)**
OBJECTIVE VALUE	16.500316	16.500313	NA	16.49418566	NA
SOLVER STATUS	NORMAL COMPLETION	NORMAL COMPLETION	NORMAL COMPLETION	NORMAL COMPLETION	ITERATION INTERRUPT
MODEL STATUS	INTEGER SOLUTION	INTEGER SOLUTION	INFEASIBLE - NO SOLUTION	INTEGER SOLUTION	NO SOLUTION RETURNED
Additional Solvers chosen by NEOS	CONOPT 3 (NLP)	XPRESS (MIP) CONOPT 3 (NLP)	-	ILOG CPLEX (LP) MINOS (NLP)	-
Iteration Count	246/10000	318/10000	47/10000	0/10000	0/10000
Resource Usage	0.340/1000.000	0.370/1000.000	62.110/1000.000	40.000/1000.000	10.336/1000.000
Compilation Time	0.037 SECONDS	0.034 SECONDS	0.036 SECONDS	0.034 SECONDS	0.037 SECONDS
Generation Time	0.024 SECONDS	0.025 SECONDS	0.014 SECONDS	0.025 SECONDS	0.014 SECONDS
Execution Time	0.026 SECONDS	0.027 SECONDS	0.016 SECONDS	0.027 SECONDS	0.016 SECONDS

Table 7-2 Comparison of Solvers in NEOS Server and Local Machine (Sengupta and Pike, 2012)

Solver	**SBB (MINLP) (NEOS Server)**	**SBB (MINLP) (Local Machine)**
OBJECTIVE VALUE	16.500316	16.500316
SOLVER STATUS	NORMAL COMPLETION	NORMAL COMPLETION
MODEL STATUS	INTEGER SOLUTION	INTEGER SOLUTION
GAMS version	GAMS Rev 228 x86/Linux	GAMS Rev 232 WIN-VIS 23.2.1 x86/MS Windows
Additional Solvers chosen by NEOS	CONOPT 3 (NLP)	CONOPT
Iteration Count	246/10000	214/ 2000000000
Resource Usage	0.340/1000.000	0.359/1000.000
Compilation Time	0.037 SECONDS	0.015 SECONDS
Generation Time	0.024 SECONDS	0.063 SECONDS
Execution Time	0.026 SECONDS	0.063 SECONDS

This paper (Vigerske and Gleixnerscip, 2016) described the extensions that were added to the constraint integer programming framework, SCIP, to allow it to solve (convex and nonconvex) mixed-integer nonlinear programs to global optimality with SCIP 3.1 (released in 2014). SCIP's implementations of optimization-based bound tightening (OBBT), branching rules, and primal heuristics for MINLP are centered around an expression graph representation of nonlinear constraints that allowed for bound tightening, detection of convex sub-expressions, and reformulation that are necessary to compute and update a linear outer-approximation based on convex over- and underestimation of nonconvex functions. The combination of discrete decisions, nonlinearity, and possible nonconvexity of the nonlinear functions in MINLP combines the areas of mixed-integer linear programming, nonlinear programming, and global optimization into a single problem class. Linear and convex smooth nonlinear programs are solvable in polynomial time in theory and very efficiently in practice, nonconvexities from discrete variables or nonconvex nonlinear functions lead to problems that are NP-hard in theory and computationally demanding in practice.

In this article (Vigerske and Gleixnerscip, 2016) the results of impact of several SCIP components on the MINLP solving performance show that disabling any of the investigated components leads to a decrease in the number of solved instances that indicates that the default settings are reasonable. The design and algorithmic features were evaluated for the impact of the individual components on its overall computational performance using the public benchmark set MINLPLib2. SCIP is actively developed and further improvements that have been made after the release of version 3.1 were not included in this paper.

Some Unique Studies for MINLP

In the following section, a number of unique studies and evaluations for global optimization algorithms and applications are described. They include connecting industrial flowsheeting simulators with MINLP solvers, determining process sheet configurations, and integrating simultaneous flow sheet optimization and heat integration, among others. Many use GAMS as the source for MINLP solvers.

Flowsheeting simulator ChemCAD was linked to the stochastic Molecular-Inspired Parallel Tempering (MIPT) algorithm for a toolbox for the systematic process retrofit of complex chemical processes by Otte, Lorenz and Repke, 2016. The flowsheeting simulator and the programming software Matlab were connected using the OPC (OLE for Process Control) standard as communication platform for data exchange and communication between Matlab and ChemCAD. The toolbox acts both as a carrier of information and for the control of ChemCAD. New optimization values (decision variables) from the MIPT algorithm are used in Matlab to execute the simulation. After the simulation is completed, the toolbox reads values from ChemCAD to optimize the objective function and satisfy constraints. The methodology was used on the retrofit of the separation of cyclohexane, benzene, toluene, and o-xylene to determine the optimal operating conditions for three given cases. The result was a set of new operating points for the process, the energy cost for the separation, and information about whether or not the demand can be fulfilled.

The minimum vapor duty requirement was used as the objective function for each distillation column configuration used in petroleum crude distillation subject to material balance constraints by Nallasivam, et al., 2016, and the Underwood equation instead of stage calculations. A NLP problem was developed for any non-azeotropic n-component separation problem using n-1 columns, and it includes configurations with and without thermal coupling. The global optimum was determined based rank-list of all possible basic and thermally coupled distillation configurations with respect to their total minimum vapor duty requirements. The optimization problem was formulated in MATLAB and called the GAMS/BARON optimization solver through the GAMS/MATLAB interface. The optimization methodology was tested with 6,28 candidate configurations located having 2,125 points with local optima and 1,625 infeasible points. Ranking the local optimum gave the global optimum. Other evaluations were performed with tighter constraints to obtain improved algorithm performance.

A simulation of a stand-alone chemicals' facility was described with main products of aromatics and allowable by- products of gasoline, liquefied petroleum gas, and electricity using natural gas a feedstock by Niziolekc, Onel, and Floudas 2015. Mass and energy balances were developed for the process units and linked together along with an economic model describing the profit based on the net present value. A mixed integer nonlinear optimization (MINLP) model was formulated and solved using a branch-and-bound global optimization algorithm to determine the optimal process topology. The mixed-integer linear relaxation was solved using CPLEX96 to determine the lower bound of the model and CONOPT was used for nonlinear optimization. The analysis was used to examine forty (40) distinct case studies across two sets of cost parameters.

A unit-specific event-based continuous-time MINLP formulation was described by Li, J, X. Xiao and C. A. Floudas, 2016 for the integrated treatment of recipe, blending, and scheduling of gasoline blending and order delivery operations. Operational features included non-identical parallel blenders, constant blending rate, minimum blend length and amount, blender transition times, multipurpose product tanks, changeovers, piecewise constant profiles for blend component qualities and feed rates, and penalty for order delivery. A hybrid global optimization was used for non-convexities in constant blending rates. Fourteen examples were solved to be 1% global optimality within modest computational effort.

A distillation configuration to have the total installation and operating costs be a minimum is described by Nallasivam, et. al., 2016, using a global minimization algorithm. For general multicomponent distillation problems, the search space is limited to distillation configurations that use exactly (n – 1) distillation columns to separate an ideal or near- ideal multicomponent mixture into n product streams. Any feasible basic distillation configuration is represented by a unique 0–1 upper triangular matrix in the matrix method and mathematical constraints ensure that only matrices corresponding to feasible basic distillation configurations that represent Underwood's equations are included in the search space. GAMS/BARON was used to guarantee global optimality with the formulation using nonlinear functions such as bilinear, fractional, or logarithmic functions and the search space being compact. The optimization problem was formulated in MATLAB and called the GAMS/BARON optimization solver through the GAMS/ MATLAB interface. A heavy crude oil distillation example was used to obtain a global optimization-based rank-list of all configurations with respect to their minimum total vapor duty requirements.

A general modelling framework was described by Grimstada, Fossa, Heddleb, and Woodman, 2016, for optimization of multiphase flow networks with discrete decision variables. They used graph-based models for oil and gas networks; spline-based surrogate models (proprietary (black-box) simulators, explicit model equations and look-up tables) to represent the nonlinear parts of the system that decouples the solver from the process simulator; and a global branch-and-bound based MINLP solver, CENSO (Convex ENvelopes for Spline Optimization), that exploits nonlinearities being described by splines and the structural properties of oil and gas networks. Case studies included three realistic production optimization cases from two BP operated subsea production systems.

Simulation-based simultaneous optimization and heat integration approach is described by Chen, et. al., 2015, linking a process simulator, Aspen Plus, a heat integration module (GAMS LP to minimize the total utility cost) and a derivative-free optimizer (Covariance Matrix Adaptation Evolutionary Strategy (CMA- ES) that is a global search optimization method suitable for difficult nonlinear nonconvex problems in continuous domains. The capabilities are demonstrated with three industrial-scale cases: a methanol production process (with a recycle stream), a separation process for benzene, and a super- critical pulverized coal power plant with post-combustion carbon capture and compression.

Data-driven and nonlinear models are described by Li, J., et. al., 2016, that used predict product yields and properties in production units of a large refinery- petrochemical complex. including crude distillation and vacuum distillation units, hydrocracking units, catalytic cracking units, ethylene-cracking units and other units. Yield and property prediction models for the crude distillation and vacuum distillation units are based on crude assay data. Binary variables denote different operation modes for several production units, or parallel production units. The planning model is a non-convex mixed integer nonlinear optimization problem that is solved using Excel linked to GAMS/ ANTIGONE solver. Several large-scale industrial examples are solved to illustrate the efficiency of the models and global optimization.

A process synthesis and global optimization framework was described by Onel, et. al. 2015 for the production of liquid fuels and olefins from biomass and natural gas used a superstructure with multiple conversion and production technologies and simultaneous heat, power, and water integration. A nonconvex mixed-integer nonlinear optimization (MINLP) was solved with a mixed-integer linear (MILP) model using CPLEX79 and a NLP using CONOPT8. The process superstructure consisted of: biomass handling and gasification, natural gas conversion, synthesis gas cleaning, hydrocarbon production, hydrocarbon upgrading, olefins purification, and a wastewater treatment network. The objective function was the summation of costs from the required feedstocks: natural gas, biomass, freshwater, and butanes and from electricity. Sixteen distinct case studies examined the capabilities of the model, investigate trade-offs of different scales and different product ratios.

A Lipschitz Global Optimizer (LGO) solver suite for constrained nonlinear global and local optimization was described by Pintér, et. al., 2016 that can solve models with continuous structure without requiring higher order, gradient, Hessian information) and its operations are based on model function values. LGO is suitable for a broad range of model calibration problems, including completely "black box" models, in addition to standard (analytically defined) models. LGO is available for use with a range of compiler platforms (C/C++/C#, Fortran 77/90/95), with seamless links to several optimization modeling languages: AMPL, GAMS, MPL, Excel, Maple, Mathematica, and MATLAB. The analytical formulation of a nonlinear regression model is outlined for an optimization problem objective function for an application study a scientific instrument, installed on-board of the International Space Station to study the Sun's effect on the Earth's atmosphere. Details are provided in the article.

Simplicial global optimization focuses on deterministic covering methods for global optimization by partitioning the feasible region by simplices as described by Paulavicˇius, and Žilinskas, 2014. A simplex is a polyhedron in a multidimensional space, which has the minimal number of vertices. The feasible region defined by linear constraints are covered by simplices. The objective function at all vertices of partitions are used to evaluate subregions. Several algorithms using this method are described, and these algorithms are evaluated using a number of classical test optimization problems. All optimization problems that were solved had linear constraints, a requirement for the algorithms to locate the global optimum.

Air Liquide operates an industrial gas pipeline network connecting air separation plants to customers of industrial gases with three pipelines: one gaseous oxygen and two gaseous nitrogen pipelines (Puranik, et. al., 2016). There are four air separation plants, each connected to at least one of the pipelines that produce high pressure, low pressure and medium pressure gaseous oxygen and nitrogen as well as liquid oxygen, liquid nitrogen and liquid argon. Gaseous products can also be bought from two competitor plants connected to the network. The demand for gaseous products can be partially satisfied through the vaporization of liquefied gases from storage. The model to describe the network uses a utilize regression-based approximate models based on historic plant data and uses a rolling horizon basis for a single time step after the uncertain demands and electricity prices are revealed. The objective is to minimize the total cost of supply to all the customers. The cost includes the total cost of production in every plant, the cost of buying gas from competitors and the cost of vaporizing gases from the liquid storage if required. The optimization model is nonconvex, necessitating the use of global optimization techniques. Results represent five different instantiations of uncertain parameter values, including atmospheric conditions, electricity prices as well as the prices of gas and liquid products. The MINLP model has 368 equality and 463 inequality constraints. There are 589 continuous and 136 binary variables in the model. BARON 15.9 was the only solver that can currently solve all five cases without any solver errors or incorrect infeasibility claims. Using MILP relaxations was required for BARON to solve the problems studied in realistic computing times.

The production of liquid transportation fuels proceeds through a synthesis gas (syngas) intermediate that can be directed to either the Fischer–Tropsch refining or methanol conversion. A process synthesis framework for a WTL refinery was developed and included: municipal solid waste gasification with/without recycle gas, syngas conversion via Fischer–Tropsch (FT) refining or methanol synthesis, methanol conversion via methanol-to-gasoline (MTG) or methanol-to-olefins (MTO), hydrocarbon upgrading via ZSM-5 zeolite catalysis, olefin oligomerization, or carbon number fractionation (Niziolek, et.al. 2015). The major liquid fuels products from the refinery include gasoline, diesel, and jet fuel, whereas liquefied petroleum gas (LPG) and electricity were to be sold as byproducts. The objective function was minimized to deter- mine the lowest cost of a WTL refinery includes the feedstock cost, CO_2 sequestration cost, levelized investment cost, electricity cost, and profit obtained from the sale of byproduct LPG. A large-scale nonconvex mixed-integer nonlinear optimization (MINLP) model was used determine the optimal process topology for liquid fuels production from many topological alternatives. A rigorous global optimization branch-and-bound strategy was employed to guarantee the global optimum objective function was determined. A mixed-integer linear relaxation was solved at each node via CPLEX. At each initial point, the binary variables were fixed, and CONOPT was called to solve the nonlinear optimization model (NLP). Twelve case studies illustrated the application of the MINLP model, and they included three sets that examined the production of different ratios of products: an unrestricted fuel output, maximization of diesel and the optimal process topologies.

This collection of papers on global optimization (Liberti and Nelson, 2006) describes details used in global optimization: symbolic manipulation algorithms, techniques for algebraic transformations, and efficient global optimization heuristics and metaheuristics for nonconvex constrained optimization problems. They include new global optimization methods, implementation of existing solvers and guidelines about building new global optimization software.

Crude oil scheduling with demand uncertainty for a typical marine-access refinery (Li, Misener and Floudas, 2012) has crude oil unloading, storage and processing that involves offshore buoy mooring stations for crude unloading and onshore facilities for crude unloading and processing. Different types of crudes can be blended in crude storage tanks. After blending, the streams are fed into the refinery units for processing. In crude oil scheduling operations, uncertainties are in demand fluctuations from ship arrivals, crude quality specifications, and some economic coefficients that can be described using discrete or continuous distributions. An example with an off-shore pipeline, five storage tanks, and two process units with a specified scheduling horizon and nominal demands was solved with a branch and bound global optimization algorithm using GAMS 22.6/CPLEX 11.0 to within 1% of global optimality. This new approach converted demand equality constraints to inequalities, and the branch and bound global optimization algorithm was extended to solve the deterministic robust counter-part optimization model. The computational results show that the generated schedule is more robust than the nominal schedule.

Random search, adaptive search, Markovian algorithms, population algorithms are defined by Schaffler, 2012. The best-known Markovian algorithm is simulated annealing. Population algorithms keep a set (the so-called population) of feasible points as a base to generate new points (by random). The set of points evolves by occasionally replacing old by newly generated members according to function values. A special type of population algorithms became popular under the name Genetic Algorithms. Similar population algorithms became known under the names of evolutionary programming, genetic programming, and memetic programming. "The lack of theoretical foundations and consequently of theoretical analysis of this type of algorithms is usually compensated by an exuberant creativity in finding terms from biology like evolution, genotype, selection, reproduction, recombination, chromosomes, survivor, parents, and descendants. The fact that these algorithms are still very popular is caused by this practice. Population algorithms have a large number of parameters in general: The MATLAB Genetic Algorithm, for instance, can be adjusted by setting 26 parameters. In 1995, the biological terminology was enlarged by names like "swarming intelligence" and "cognitive consistency" from behavior research without improvement of the methods." For unconstrained global optimization, detailed mathematics and examples are given for the randomized curve of steepest descent. For constrained global minimization, an active set method is recommended using the randomized projected curve of steepest descent. The text is concluded with vector optimization and a review of probability theory.

Process synthesis and design is the selection of the topology, the flowsheet, and the operating conditions to transform a set of raw materials into products and involves discrete and continuous decisions giving rise to a mixed-integer nonlinear programming problem (MINLP) or a generalized disjunctive programming (GDP) model according to Martin, 2014. Descriptions and examples of using GAMS for optimization of an ammonia reactor, SO_2 catalytic converter, steam reforming of natural gas, and process superstructure are given, including GAMS programs.

Generalized disjunctive programming (GDP) originated with the goal of facilitating the modeling of discrete/continuous optimization problems through the use of higher-level logic constructs. This approach involves algebraic equations, disjunctions and logic propositions in the formulation of a model. Details and examples are given by Grossmann and Trespalacios, 2013.

Multiobjective Optimization

Multiobjective optimization, also called multicriteria optimization, is the simultaneous optimization of more than one objective function. The general Multiobjective Problem (MOP) is defined as in Equation 7-3:

$$\text{Optimize:} \quad F(x) = [f_1(x), f_2(x), \ldots, f_k(x)]^T \tag{7-3}$$

$$\text{Subject to:} \quad g_i(x) \geq 0 \qquad i = 1, 2, \ldots, m$$

$$h_j(x) = 0 \qquad j = 1, 2, \ldots, p$$

$$x_L \leq x \leq x_U$$

There are various methods to solve multicriteria optimization problems like utility function, hierarchical methods and goal programming (Rangaiah and Bonilla-Petriciolet 2013 and Rao, 2009). Of these, using the utility function or weighted objective method is the most commonly used. In this method, weights are assigned to the different objective functions and the sum of the weights times the objective functions is formed for a single objective function as shown in Equation 7-4.

$$\min \sum_{i=1}^{k} w_i f_i(x), \text{ where } \sum_{i=1}^{k} w_i = 1,\ w_i \geq 0, \tag{7-4}$$

The multicriteria problem can be a mixed integer nonlinear programming problem where the multiple objective functions and the constraints are non-linear, and the variables are continuous or integer. The MINLP problem in the research described below was formulated into a multi-criteria problem by maximizing the profit and the sustainability credits simultaneously.

A detailed review of multicriteria optimization in sustainable energy decision-making was given by Wang et al., 2009. Technical criteria, economic criteria, environmental criteria and social criteria were discussed in the paper along with weighted objective methods.

Multiobjective Optimization Problem Statement for a Chemical Production Complex

The statement for the optimization problem for a chemical production complex by Sengupta and Pike, 2012 is:

Optimize: Objective Function

Subject to: Constraints from plant models

The objective function is a profit function for the triple bottom line, Equation 7-5.

Triple Bottom Line = Profit - Σ Environmental Costs + Σ Sustainable (Credits – Costs) (7-5)

The profit in Equation 7-5 is described using an extended value-added economic model, Equation 7-6.

$$\text{Profit} = \Sigma \text{ Product Sales} - \Sigma \text{ Raw Material Costs} - \Sigma \text{ Energy Costs} \quad (7\text{-}6)$$

Substituting in Equation 7-5 gives the objective function used in the multicriteria optimization.

$$\text{Triple Bottom Line} = \Sigma \text{ Product Sales} - \Sigma \text{ Raw Material Costs} - \Sigma \text{ Energy Costs} - \Sigma \text{ Environmental Costs} + \Sigma \text{ Sustainable (Credits} - \text{Costs)}$$

The constraint equations describe relationship among variables and parameters in the processes and plants. Equality constraints are material and energy balances, chemical reaction rates, thermodynamic equilibrium relations and others. Inequality constraints are availability of raw materials, demand for products, capacities of process units and others.

The objective of multicriteria optimization is to find optimal solutions that maximize industry' profits and minimize costs to society. This multicriteria optimization problem can be stated as in terms of industry's profit, P, and society's sustainable credits/costs, S; and these two objectives are given by Equation 7-7.

Max: $P = \Sigma$ Product Sales - Σ Economic Costs - Σ Environmental Costs (7-7)
$S = \Sigma$ Sustainable (Credits – Costs)

Subject to: Multi-plant material and energy balances,
product demand, raw material availability, plant capacities

To locate Pareto optimal solutions, multi-criteria optimization problems are converted a single criterion by applying weights to each objective and optimizing the sum of the weighted objectives as shown in Equation 52 where $w_1 + w_2 = 1$.

Max: $w_1P + w_2 S = w_1 P + (1 - w_1) S$ (7-8)

Subject to: Multi-plant material and energy balances,

product demand, raw material availability, plant capacities

If w_1 is 0, then only industry profits are considered, and no sustainable costs/credits are included. If $w_1 = 1$ the only sustainable costs/credits are evaluated at the Pareto optimum. With $w_1 = 0.5$ industry profits and sustainable cost/credits are weighted equally. Results are summarized in Figure 35 for the chemical production complex. It is another decision to determine the specific value of the weight that is acceptable to all concerned.

The Chemical Complex Analysis System was used to determine the Pareto optimal solutions for the weights using $w_1+w_2=1$ given by Equation 7-8, and these results are shown in Figure 7-2. The profits for the company are two orders of magnitude larger than the sustainable credits/costs. The sustainable credits/costs decline, and company's profits increase as the weight,

w_1, on company's profit increase. For example, when w_1=1, the optimal solution is shown in Table 7-3 for P=$1660.01 million per year and S=$-9.98 million per year. The optimal solution with w_1=0 gave P=$1193.45 million per year and S=$26.00 million per year. The points shown in Figure 7-2 are the Pareto optimal solutions for w_1 from 0 to 1.0 for increments of 0.001.

The values for w_1 equal to 0 and 1.0 and some intermediate ones are shown in Table 7-3. The optimal complex configurations of the Pareto optimal solutions for w1 from 0 to 1.0 for increment of 0.001 are shown in Table 7-3. If a process is selected, the binary variable associated with the process is 1, otherwise 0. For each process in Table 7-3, the sums of the binary variable values for the corresponding w_1 range are shown, along with the total summation of the times the process was selected. See Sengupta and Pike, 2012.

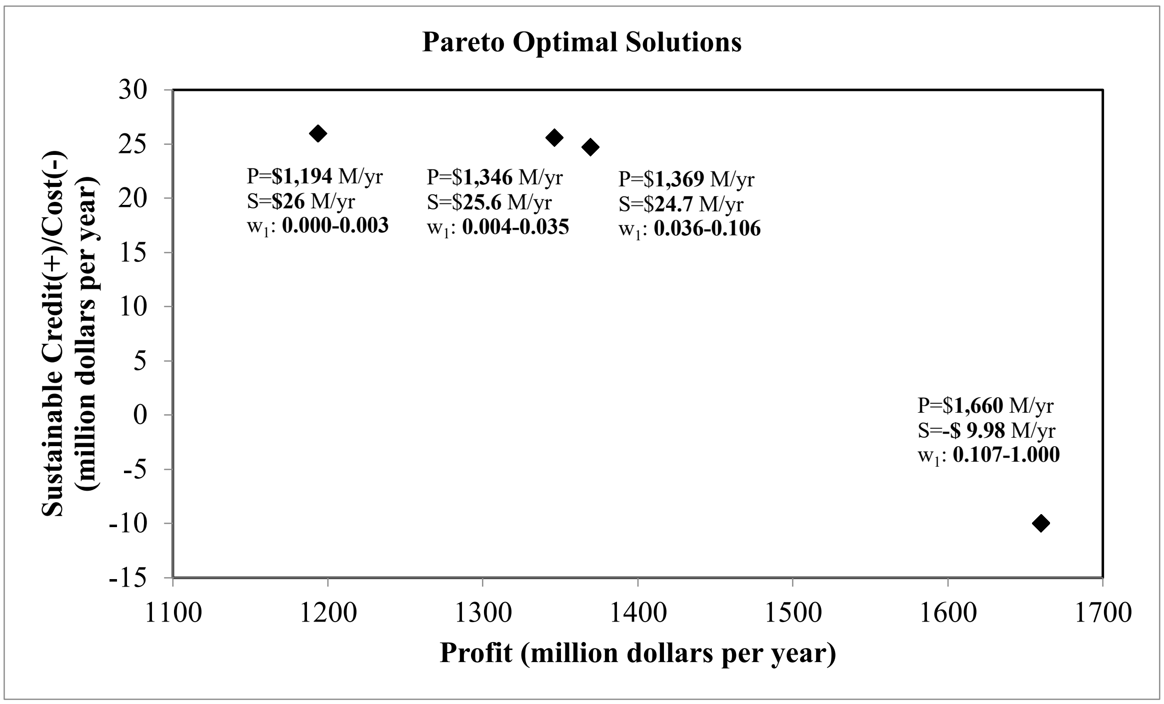

Figure 7-2 Optimal Solutions Generated by Multicriteria Optimization

Table 7-3 Values of the Pareto Optimal Solutions shown in Figure 7-2

Profit (million dollars/year)	Sustainable Credits/Costs (million dollars/year)	Weight (w_1)
1660.01	-9.98	1
1660.01	-9.98	0.894
1660.01	-9.98	0.107
1369.32	24.74	0.106
1369.32	24.74	0.036
1346.26	25.60	0.035
1346.26	25.60	0.004
1193.94	26.00	0.003
1193.45	26.00	0

Summary

Global optimization algorithms are either deterministic or stochastic methods. The most successful deterministic strategies include inner and outer approximation methods, branch and bound methods, cutting plane methods and interval bounding methods. Successful stochastic strategies include random search, genetic algorithms and simulated annealing.

Chemical process systems optimization problems frequently involve both continuous and binary variables and have the form of mixed integer nonlinear programming (MINLP) problems. The continuous variables represent the flow rates, temperature, pressures, etc., and binary variables represent the configuration of process units. These problems have been difficult to solve, and a significant amount of research has been spent developing algorithms that are effective in solving MINLP problems for the global optimum.

Deterministic optimization of a MINLP problem for a chemical process system is usually accomplished using an algorithm like the branch and bound or the inner-outer method. These algorithms solve a series of NLP problems that typically use the generalized reduced gradient method (GRG) or successive (sequential) quadratic programming (SQP). These NLP algorithms have a super-rate of convergence and locate the optimum in 2n steps for quadratic functions.

Branch and bound methods use a systematic enumeration of candidate solutions that are thought of as forming a tree with the full set of solutions at the top of the tree. The algorithm explores branches of this tree that represent subsets of the solution set. Each branch is checked against upper and lower estimated bounds on the optimal solution and branches are discarded if they cannot produce a better solution than the best one found so far by the algorithm

Nonconvex MINLPs pose additional challenges, because they contain nonconvex functions in the objective and or the constraints. Spatial branch-and-bound is the best-known method for solving nonconvex MINLP problems. Most modern MINLP solvers designed for nonconvex problems utilize a combination of the techniques. In particular, they are branch-and-bound algorithms with at least one rudimentary bound-tightening technique and a lower-bounding procedure.

The General Algebraic Modeling System (GAMS) is a high-level modeling language for mathematical programming and optimization. It was specifically designed for solving linear, nonlinear and mixed integer optimization problems. The system is especially useful with large, complex problems. GAMS is available for use on personal computers, workstations, mainframes and supercomputers.

An overview of the start-of-the-art in software for the solution of mixed integer nonlinear programs (MINLP) is given by Bussieck and Vigerske, 2014, of GAMS that describes various features of embedded and independent solvers with a concise description for each solver to provide to guide the selection of a best solver for a particular MINLP problem. Methods used by several established MINLP solvers include BARON, COCONUT, COUENNE and LindoGlobal.

References

Aarts, E. H. L. and H. M. M. Ten Eikelder, 2002, “Simulated Annealing,” *Handbook of Applied Optimization*, Pardalos, P. M. and M. G. C. Resende, Editors, Oxford University Press, New York, NY

Belotti, Pietro, Christian Kirches, Sven Leyffer, Jeff Linderoth, Jim Luedtke, and Ashutosh Mahajan, (2012) Mixed-Integer Nonlinear Optimization, Preprint ANL/MCS-P3060-1112, Mathematics and Computer Science Division, Argonne National Laboratory, November 22, 2012

Bussieck, M. R. and S. Vigerske, “MINLP Solver Software,” GAMS Report, GAMS Development Corp., 1217 Potomac St, NW Washington, DC 20007, USA (March 10, 2014)

Byrne, R. P. and I. D. L. Bogle, (2000), Global Optimization of Modular Process Flowsheets, Ind. Eng. Chem. Res. 2000, 39, 4296-4301

Chen, Y., J. C. Eslicka, I. E. Grossmann, D. C. Miller “Simultaneous process optimization and heat integration based on rigorous process simulations,” Computers and Chemical Engineering 81 (2015) 180–199

Choi, S. H. and Vasilios Manousiouthakis, 2002, Global Optimization Methods for Chemical Process Design: Deterministic and Stochastic Approaches, Korean J. Chem. Eng., 19(2), 227-232.
Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison –Wesley, New York

Gray, P. W. Hart, L. Painton, C. Phillips, M. Trahan, J. Wagner, *A Survey of Global Optimization Methods, Sandia* National Laboratories*,* Albuquerque, NM 87185 www.cs.sandia.gov/opt/**survey**/ accessed 11/11/2014

Grimstada, B, B. Fossa, R. Heddleb, and M. Woodman, “Global optimization of multiphase flow networks using spline surrogate models,” Computers and Chemical Engineering 84 (2016) 237–254

Grossman, I.E., 1990, "MINLP Optimization and Algorithms for Process Synthesis," *Foundations of Computer Aided Process Design*, J.J. Sirola, I.E. Grossman and G. Stephanopoulos, Eds., Elsevier, New York.

Grossmann, I. E., and F. Trespalacios, 2013, “Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming,” AIChE Journal, Volume 59, Issue 9, September 2013, Pages 3276–3295

Jeroslow, R. C. (1973), There Cannot be any Algorithm for Integer Programming with Quadratic Constraints. Operations Research 21(1):221-224

Jeroslow, R. C. (1974), Trivial Integer Programs Unsolvable by Branch-And-Bound, Mathematical Programming 6 (1974) 105-109.

Kronqvista Jan, D. E. Bernalb, A. Lundellc, and I. E. Grossmann, 2018 A Review and Comparison of Solvers for Convex MINLP, Optimization Online, Mathematical Optimization Society, June 5, 2018.

Li J., R. Misener and C. A. Floudas, "Scheduling of Crude Oil Operations Under Demand Uncertainty: A Robust Optimization Framework Coupled with Global Optimization," AIChE Journal Vol. 58, No. 8, August 2012

Li J., X. Xiao, C. A. Floudas, "Integrated Gasoline Blending and Order Delivery Operations: Part I. Short-Term Scheduling and Global Optimization for Single and Multi-Period Operations," Vol. 62, No. 6 AIChE Journal, p. 2043-2070 (2016).

Li, J., F. Boukouvala, X. Xiao, C. A. Floudas, B. Zhao, G. Du, Xin Su, and H. Liu, "Data-Driven Mathematical Modeling and Global Optimization Framework for Entire Petrochemical Planning Operations," doi: 10.1002/aic.15220, American Institute of Chemical Engineers, 2016

Liberti, L., M. Nelson, Eds., "Global Optimization: from Theory to Implementation," Springer Science+Business Media, Inc., 233 Spring Street, New York, NY 10013, USA, 2006

Martin, M. M., "Use of GAMS for Optimal Process Synthesis and Operation," Introduction to Software for Chemical Engineering, Chapter 11, CRC Press, Boca Raton, FL 2014.

Nallasivam, U., V. H. Shah, A. A. Shenvi, J. Huff, M. Tawarmalani and R. Agrawal, "Global Optimization of Multicomponent Distillation Configurations: 2. Enumeration based Global Minimization Algorithm," AIChE Journal, doi: 10.1002/aic.15204, © 2016 American Institute of Chemical Engineers (AIChE)

Nallasivam, U., V. H. Shah, A. A. Shenvi, J. Huff, M. Tawarmalani and R. Agrawal, "Global Optimization of Multicomponent Distillation Configurations: 2. Enumeration Based Global Minimization Algorithm," AIChE Journal, Vol. 62, No. 6, p. 2071- 2086 (June,2016).

Niziolek, A. M., O. Onur, M.M. Faruque Hasan, C. A. Floudas, "Municipal solid waste to liquid transportation fuels – Part II: Process synthesis and global optimization strategies," Computers and Chemical Engineering 74 (2015) 184–203

Niziolekc, A. M., O. Onel, and C. A. Floudas, "Production of Benzene, Toluene, and Xylenes from Natural Gas via Methanol: Process Synthesis and Global Optimization," doi:10.1002/aic.15144, © 2015 American Institute of Chemical Engineers (AIChE)

Onel, O, A. M. Niziolek, J. A. Elia, R. C. Baliban, and C. A. Floudas," Biomass and Natural Gas to Liquid Transportation Fuels and Olefins (BGTL+C2_C4): Process Synthesis and Global Optimization," Ind. Eng. Chem. Res. 2015, 54, 359−385

Otte, D. H-M. Lorenz and J. W. Repke, "A toolbox using the stochastic optimization algorithm MIPT and ChemCAD for the systematic process retrofit of complex chemical processes," Computers and Chemical Engineering 84 (2016) 371–381

Pardalos, P. M. and M. G. C. Resende, Editors, 2002, *Handbook of Applied Optimization*, Oxford University Press, New York, NY

Paulavicˇius, R. and Julius Žilinskas, Simplicial Global Optimization Springer New York Heidelberg Dordrecht London, 2014

Pike, Ralph W. 2013, *Optimization for Engineering Systems Revised*, (Kindle Edition) ASIN: B00BF2TLXO Amazon.com (2013)

Pinter, J. D., 2014, "How Difficult is Nonlinear Optimization? A Practical Solver Tuning Approach, with Illustrative Results "Optimization *Online*, Mathematical Optimization Society, June, 2014

Pintér, J. D., A. Castellazzo, M. Vola2, and G. Fasano, "Nonlinear Regression Analysis by Global Optimization: A Case Study in Space Engineering," Optimization Online Digest, March 2016

Puranik, Y., M. Kılınç, N. V. Sahinidis, T. Li, A. Gopalakrishnan, B. Besancon, and T. Roba, "Global optimization of an industrial gas network operation," AIChE Journal, doi: 10.1002/aic.15344, May 19, 2016

Rangaiah and Bonilla-Petriciolet 2013, *Multiobjective Optimization in Chemical Engineering: Developments and Applications*, Wiley, Hoboken, NJ

Rao, 2009, *Engineering Optimization Theory and Practice*, Fourth Ed., Wiley, Hoboken, NJ

Sahinidis, Nick, 2005, Global Optimization and Optimization under Uncertainty, Pan American Study Institute on Process Systems Engineering, Iguazu Falls, Argentina, August 18, 2005

Schaffler, S., Global Optimization A Stochastic Approach, Springer New York 2012

Sengupta, D and R. W. Pike, Chemicals from Biomass: Integrating Bioprocesses into Chemical Production Complexes for Sustainable Development, CRC Press, Boca Raton, FL, 2012.

Trafalis, T. B. and S. Kaspa 2002, "Artificial Neural Networks in Optimization and Applications," *Handbook of Applied Optimization*, Pardalos, P. M. and M. G. C. Resende, Editors, Oxford University Press, New York, NY

Trespalacios, F., and I. E. Grossmann, (2014) "Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods," Chem. Ing. Tech. Vol. 86, No. 7, 991–1012, 2014

Vigerske, S. and A. Gleixnerscip, "Global Optimization of Mixed-Integer Nonlinear Programs in a Branch-and-Cut Framework," ZIB Report 16-24, Zuse Institute Berlin, Takustrasse 7 D-14195 Berlin-Dahlem Germany, May 2016

Wang, W-U, et al, 2009, Review on multi-criteria decision analysis aid in sustainable energy decision-making, Renewable and Sustainable Energy Reviews 13(9)