CHAPTER 11

ADVANCES IN LOGIC-BASED OPTIMIZATION APPROACHES TO PROCESS INTEGRATION AND SUPPLY CHAIN MANAGEMENT

Ignacio E. Grossmann

Department of Chemical Engineering

Carnegie Mellon University

Pittsburgh, PA 15213, U.S.A.

Abstract. Optimization as an enabling technology has been one of the big success stories in process systems engineering. In this paper we present a review on recent research work in the area of logic-based discrete/continuous optimization. In particular, recent advances are presented in the modeling and solution of nonlinear mixed-integer and generalized disjunctive programming, global optimization and constraint programming. The impact of these techniques is illustrated with several examples in the areas of process integration and supply chain management..

11.1 Introduction

Our objective in this chapter is to provide an overview of new developments in discrete/continuous optimization with applications to process integration and supply chain management problems. The emphasis is on logic-based optimization which is becoming a new promising area in process systems engineering.

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

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

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 $g(x,y) \leq 0$ 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.

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 Integration (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. An extensive review of optimization models for process integration can be found in Grossmann et al. (1999). In contrast, Supply Chain Management problems tend to be dominated by linear models, LP and MILP, for planning and scheduling (see Grossmann et al. 2002 for a review). Finally, 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).

In the next section 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.

11.2 Logic-Based Discrete and Continuous Optimization

11.2.1 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,

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

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, $0 \le y \le 1$. 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:

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

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.

11.3 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 (2000).

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:

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

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 $g(x) \leq 0$ hold regardless of the discrete choice, and $h_{jk}(x) \leq 0$ 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},

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

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.

11.3.1 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,

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

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).

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

where U is a valid upper bound for x and v. For computational reasons, the nonlinear inequality is written as $(\lambda_{jk} + \varepsilon) h_{jk}(v_{jk} / (\lambda_{jk} + \varepsilon)) \le 0$ 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).

11.4 Solution Algorithms for GDP

11.4.1. 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.

11.4.2 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, although this leads to a weaker relaxation (Grossmann and Lee, 2003). 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 QP, is solved:

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

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,

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

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

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

where $\beta^T x \leq b$ 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.

11.4.3 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:

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

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 | Y^l_{jk} = $true$ }, then one can define the following disjunctive OA master problem:

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

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 $Ay \leq a$, 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).

11.4.4 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:

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

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.

11.5 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 (2002) 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 (2000b) developed a generalized branch-and-cut (GBC) algorithm, and showed that their earlier decomposition algorithm (Kesavan and Barton, 2000a) 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.

11.5.1 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.

11.6 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.

11.7 Examples in process integration

11.7.1 Synthesis of Separation System

This problem, a joint collaboration with BP (Lee et al., 2003), deals with the synthesis of a separation system of an ethylene plant in which the mixture to be separated includes hydrogen, methane, ethane, ethylene, propane, propylene, and C4s, C5s, and C6s. For each potential separation task 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, which includes 53 separation tasks, is shown in Fig. 11.1. 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 that is shown in Fig. 11.2 included a dephlegmator and a physical absorber, and one less distillation column, achieving a $20 million reduction in the cost, largely from reduced refrigeration.

11.7.2 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).

For a ten process instance (see Fig. 11.3) 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.

11.7.3 Wastewater treatment network

This example corresponds to a synthesis problem of a distributed wastewater multicomponent network, which is taken from 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. 11.4 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 (2003) 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. 11.5.

11.8 Examples in Supply Chain Management

11.8.1 Hydrocarbon field infrastructure planning

In this example we consider the design, planning and scheduling of an offshore oilfield infrastructure over a planning horizon of 6 years divided into 24 quarterly periods where decisions need to be made (see Van den Heever and Grossmann, 2000). The infrastructure under consideration consists of one Production Platform (PP), 2 Well Platforms (WP), 25 wells and connecting pipelines (see Figure 11.6). Each oilfield (F) consists of a number of reservoirs (R), while each reservoir in turn contains a number of potential locations for wells (W) to be drilled. Design decisions involve the capacities of the PPs and WPs, as well as decisions regarding which WPs to install over the whole operating horizon. Planning decisions involve the production profiles in each period, as well as decisions regarding when to install PP and WPs included in the design, while scheduling decisions involve the selection and timing of drilling of the wells. This leads to an MINLP model with 9744 constraints, 5953 continuous variables, and 700 0-1 variables. An attempt to solve this model with a commercial package such as GAMS (Brooke et al., 1998) (using DICOPT (Viswanathan and Grossmann, 1990)) with CPLEX 6.6 (ILOG, 2000) for the MILPs and CONOPT2 (Drud, 1994) for the NLPs on an HP9000/C110 workstation), results in a solution time of 19386 CPU s.

To overcome this long solution time, Van den Heever and Grossmann (2000) developed an iterative aggregation/disaggregation algorithm which solved the model in 1423 CPU seconds. This algorithm combines the concepts of bilevel decomposition, time aggregation and logic-based methods. The application of this method led to an order of magnitude reduction in solution time, and produced an optimal net present value of $ 68 million. Figure 11.7 shows the total oil production over the 6 year horizon, while Table 11.1 shows the optimal investment plan obtained. Note that only 9 of the 25 wells were chosen in the end. This solution resulted in savings in the order of millions of dollars compared to the heuristic method used in the oilfield industry that specify almost all the wells being drilled.

11.8.2 Supply Chain Problem

The example problem in Figure 11.8, a multisite continuous production facility for multiple products, was solved by Bok et al. (2000) over a 7 day horizon to illustrate the performance of their model in three cases: (1) no intermittent deliveries of raw materials without product changeovers, (2) intermittent deliveries without changeovers, (3) intermittent deliveries with changeovers. It is obvious that case 3 is the most rigorous and detailed. Cases 2 and 3 can be obtained by relaxing the discrete nature of case 3. In the case of intermittent deliveries, the minimum time interval between successive deliveries is assumed to be 2 days regardless of the chemicals or the sites. The problem was modeled using the GAMS modeling language and solved in the full space using the CPLEX solver on a HP 9000/7000. The optimization results for this example 1 are as follows. In case 1, 2034 variables, 1665 constraints are required and no 0-1 variable is needed because there are no intermittent deliveries nor any changeovers. The problem was solved in only 2sec CPU time. The more rigorous the model (cases 2 or 3), the larger the number of 0-1 variables that is required. This in turn results in more computation time for the optimization. Case 2 for changeovers required 224 0-1 variables, 1810 continuous variables, and 1665 constraints solving in 4 min CPU. Case 3, that includes changeovers and intermittent supplies involved 392 0-1 variables, 1642 continuous variables, and 1665 constraints solving in 8 min CPU. Figure 11.9 shows the optimization results for case 3 that considers the intermittent deliveries and changeovers. Bok et al. (2000) proposed a decomposition algorithm in order to be able to solve larger problems.

11.8.2 Scheduling of Batch Plants

Consider the batch scheduling problem that is given through the State-Task-Network (STN) shown in Fig. 10 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 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 11 is guaranteed to be the global optimum solution.

11.9 Conclusions

Mixed-integer optimization techniques (MILP, MINLP) have proved to be essential in modeling process integration problems (process synthesis and design) as well as supply chain management problems (planning and scheduling). 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. The power and scope of the techniques was demonstrated on a variety of Process Integration and Supply Chain Management problems.

Acknowledgments

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

11.10 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.

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.

Baptiste, P. and Le Pape, C.,"Disjunctive Constraints for Manufacturing Scheduling: Principles and Extensions," *International Journal of Computer Integrated Manufacturing*, 9(4), 306-3410 (1996).

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.

Bergamini, M. L., P. Aguirre, and I.E. Grossmann, "Logic Based Outer Approximation for Global Optimization of Synthesis of Process Networks," submitted for publication, 2004.

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

Bockmayr, A.; Pisaruk, N. Detecting Infeasibility and Generating Cuts for MIP Using CP. In Proceedings of CP-AI-OR 2003, 24-34, 2003.

Bok, J-K, I.E. Grossmann and S. Park, "Supply Chain Optimization in Continuous Flexible Process Networks", *I&EC Res.* **39**, 1279-1290 (2000).

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.

Brooke A., D. Kendrick, A. Meeraus and R. Raman, GAMS Language Guide, GAMS Development Corporation, Washington, DC (1998).

Caseau, Y. and F. Laburthe, " Improved CLP Scheduling with Task Intervals." In Proceedings of 11th International Conference on Logic Programming, 1994

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

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.

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

Fourer, R., D.M. Gay and B.W. Kernighan, "AMPL: A Modeling Language for Mathematical Programming," Duxbury Press, Belmont, CA (1992).

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*, 3, 227-252 (2002).

Grossmann, I.E. and S. Lee, "Generalized Disjunctive Programming: Nonlinear Convex Hull Relaxation and Algorithms", *Computational Optimization and Applications* **26**, 83-100(2003).

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., S.A. van den Heever and I. Harjunkoski, "Discrete Optimization Methods and their Role in the Integration of Planning and Scheduling," AIChE Sympsium Series No. 326, Vol. 98, pp.150-168 (2002)

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

Heipcke, S. An Example of Integrating Constraint Programming and Mathematical Programming. Electronic Notes in Discrete Mathematics, 1(1), 1999.

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

Hentenryck, P.V. Constraint and Integer Programming in OPL. *INFORMS Journal on Computing*, 14(4), 345-372 (2002).

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, New York, 2000.

Hooker, J.N. Logic, Optimization, and Constraint Programming. *INFORMS Journal on Computing*, 4(4), 295-321 (2002).

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).

Jain, V.; Grossmann, I.E. Algorithms for Hybrid MILP/CP Model for a Class of Optimization Problems. INFORMS Journal in Computing, 13, 258-276 (2001).

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, 2-23.

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*, 96(323):458-461, (2000a).

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

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.

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, "Global Optimization of Nonlinear Generalized Disjunctive Programming with Bilinear Equality Constraints: Applications to Process Networks," *Computers and Chemical Engineering* 27,1557-1575 (2003).

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.

Le Pape, C. Implementation of Resource constraints in ILOG SCHEDULE: A Library for the Development of Constrained-Based Scheduling Systems. *Intelligent Systems Engineering*, 3(2), 55-66 (1998).

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.

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.

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

Nuijten, W.P.M. Time and Resource Constrained Scheduling: A Constraint Satisfaction Approach. PhD Thesis, Eindhoven University of Technology, 1994.

Papageorgiou, L.G.; Pantelides, C.C. Optimal Campaign Planning/Scheduling of Multipurpose Batch/Semicontinuous Plants. 2. Mathematical Decomposition Approach. *Ind. Eng. Chem. Res*., 35, 510-529 (1996).

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.

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.

Rodosek, R.; Wallace, M.G.; Hajian, M.T. A New Approach to Integrating Mixed Integer Programming and Constraint Logic Programming. *Annals of Operations Research*, 86, 63-87 (1999).

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.

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

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, S.F.; Cheng, C-C. Slack-based Heuristics for Constrained Satisfaction Scheduling. Proceedings of 11th National Conference on Artificial Intelligence, 1993.

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.

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

Tawarmalani, M. and N. V. Sahinidis, Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, Kluwer Academic Publishers, Dordrecht, Vol. 65 in "Nonconvex Optimization And Its Applications" series (2002)

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.

Van den Heever, S.A., and I.E. Grossmann, "An Iterative Aggregation/Disaggregation Approach for the Solution of a Mixed Integer Nonlinear Oilfield Infrastructure Planning Model," *I&EC Res.*39, 1955-1971 (2000).

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.

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

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.

Table 11.1. The optimal investment plan

Item		Period invested
PP		Jan. 1999
WP1		Jan. 1999
Reservoir	**Well**	
2	4	Jan. 1999
3	1	Jan. 1999
5	3	Jan. 1999
4	2	Apr. 1999
7	1	Jul. 1999
6	2	Oct. 1999
1	2	Jan. 2000
9	2	Jan. 2000
10	1	Jan. 2000

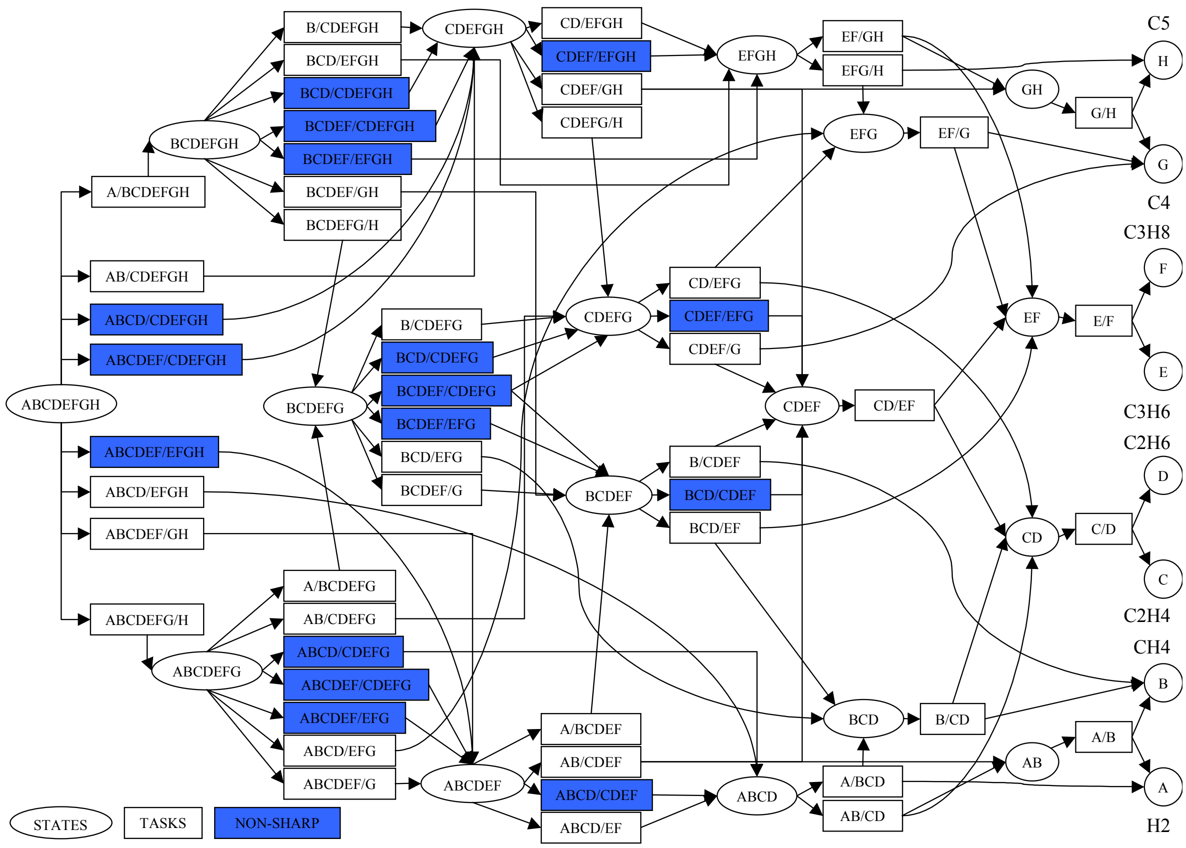


Figure 11.1: Superstructure of Separation of Ethylene Plant

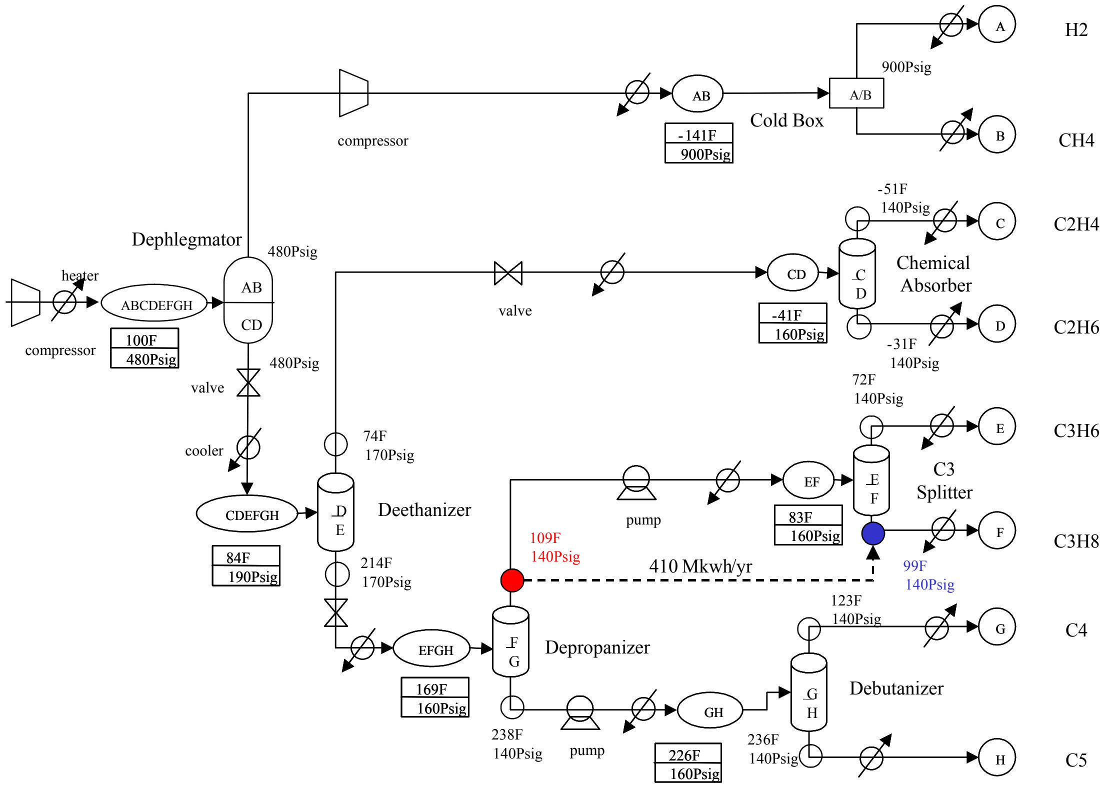


Figure 11.2:.Optimal Structure of Ethylene Plant

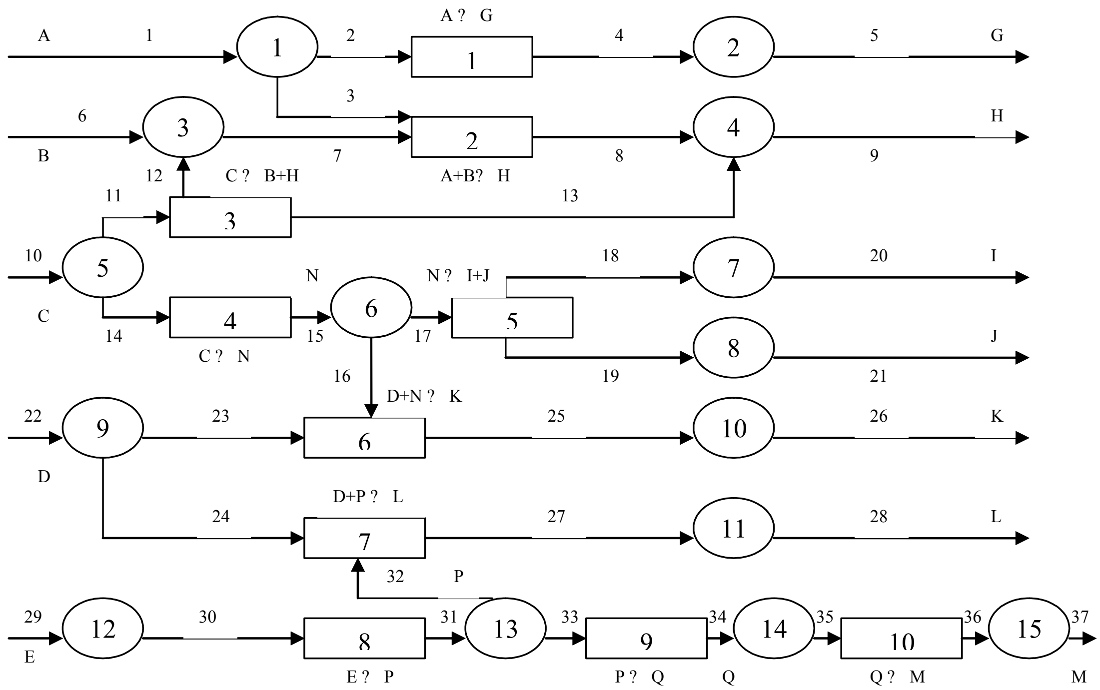


Fig. 11.3: Process network for retrofit planning

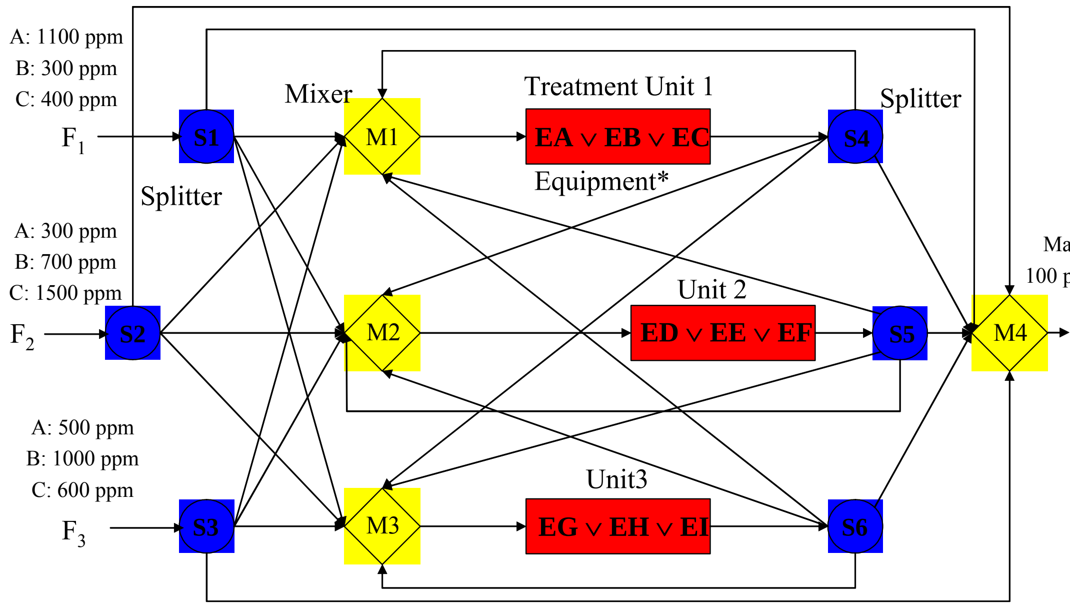


Figure 11.4:. Superstructure water treatment plant.

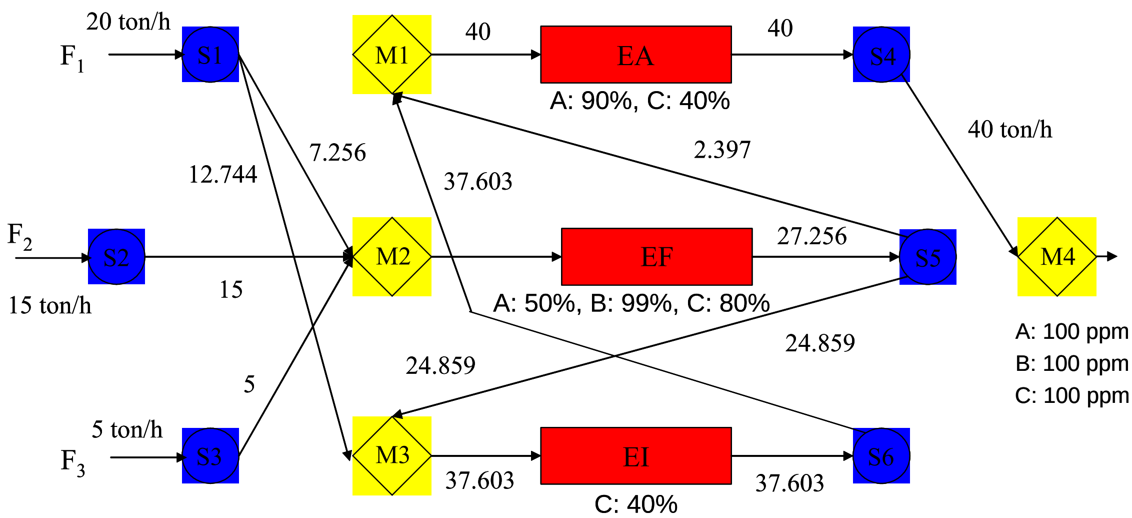


Figure 11.5: Optimal wastewater treatment plant.

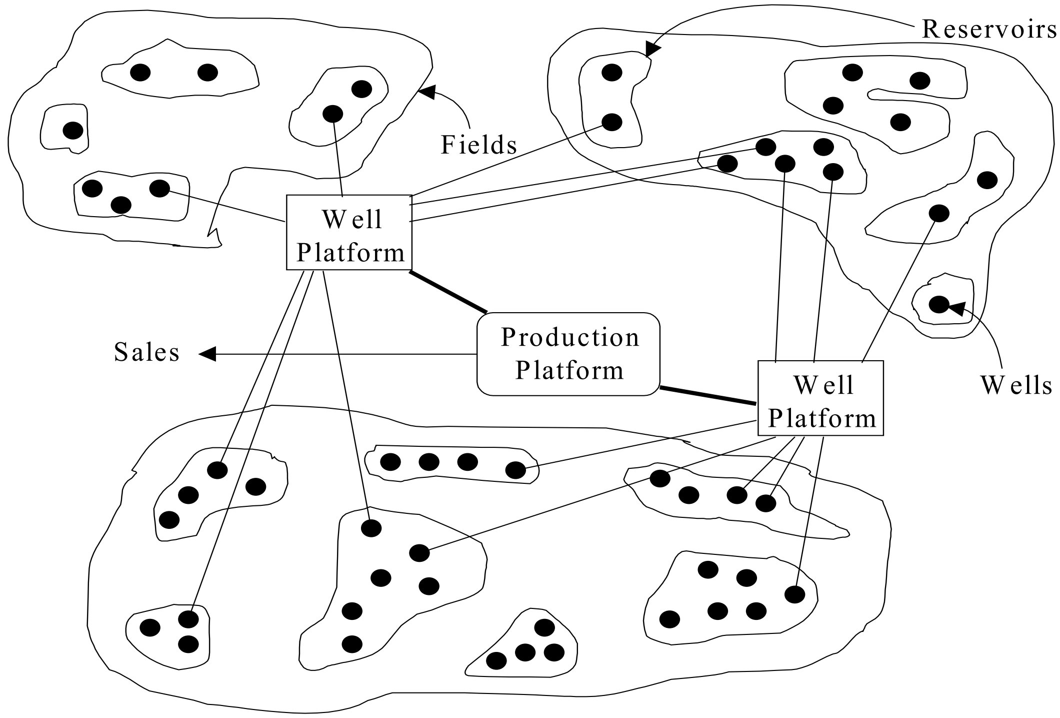


Figure 11.6: Configuration of fields, well platforms and production platforms

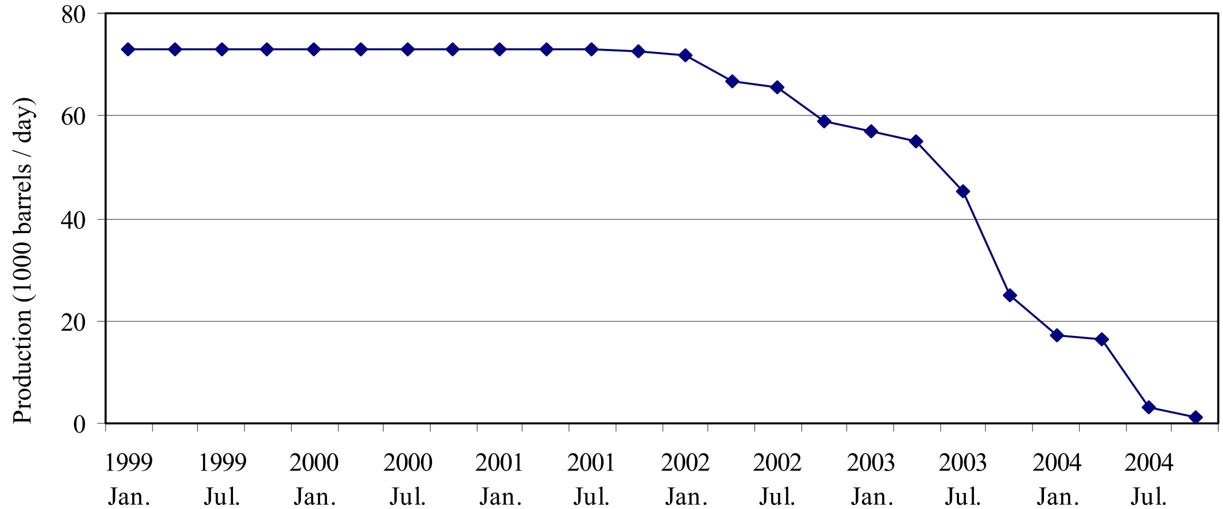


Figure 11.7: Production profile over 6 year horizon

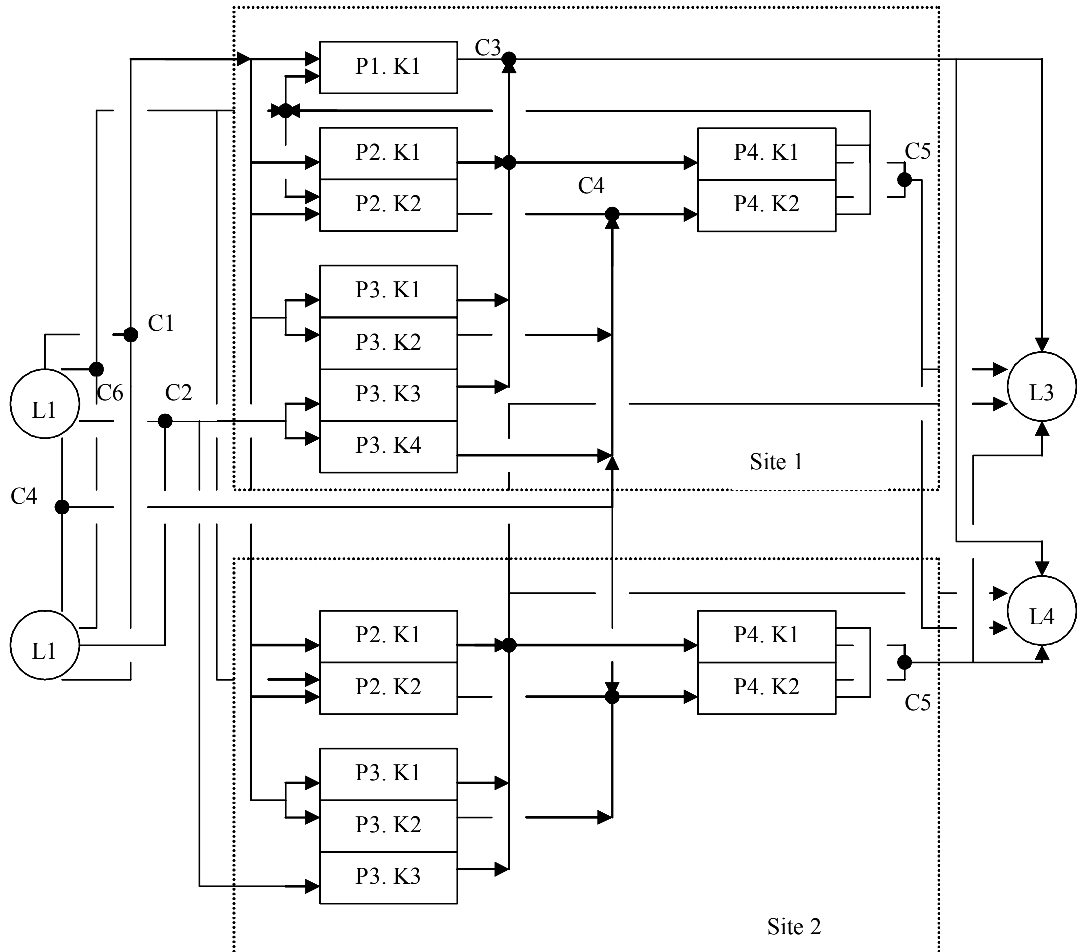
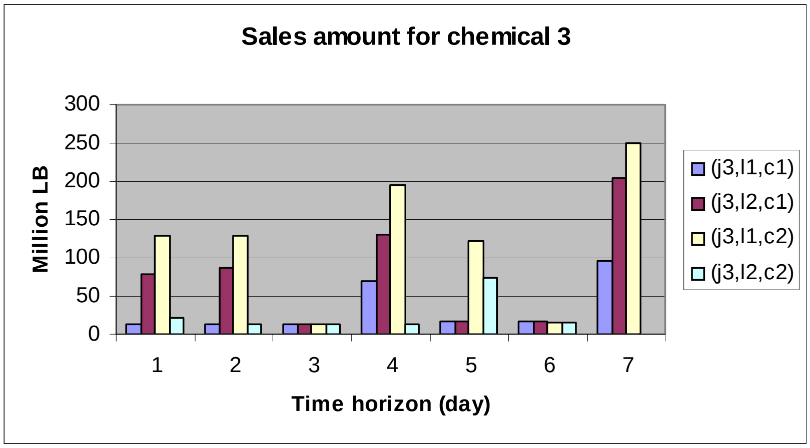
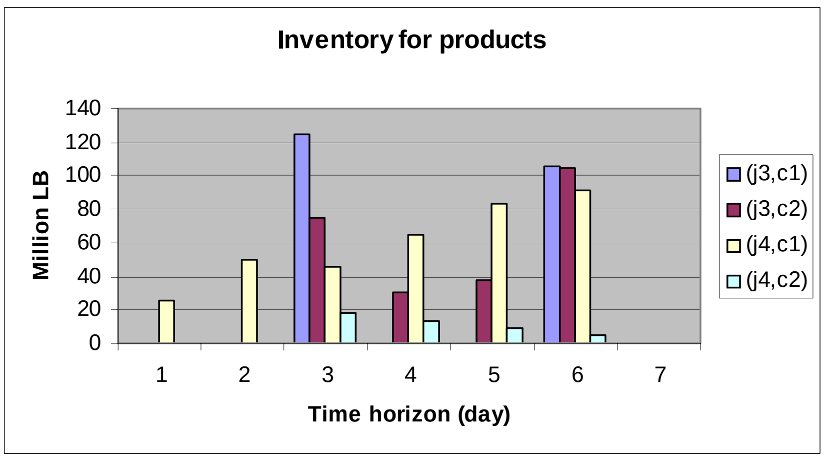


Fig. 11.8. Multisite production facility for supply chain problem





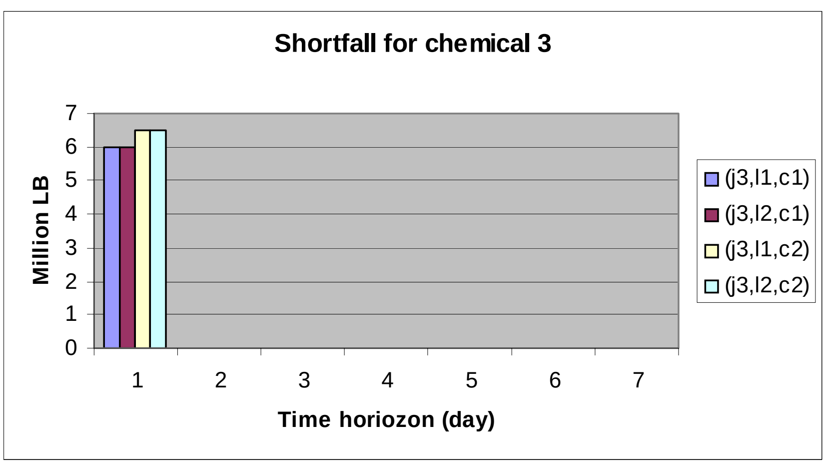


Fig. 11.9. Results for sales, inventory and shortfalls in supply chain problem.

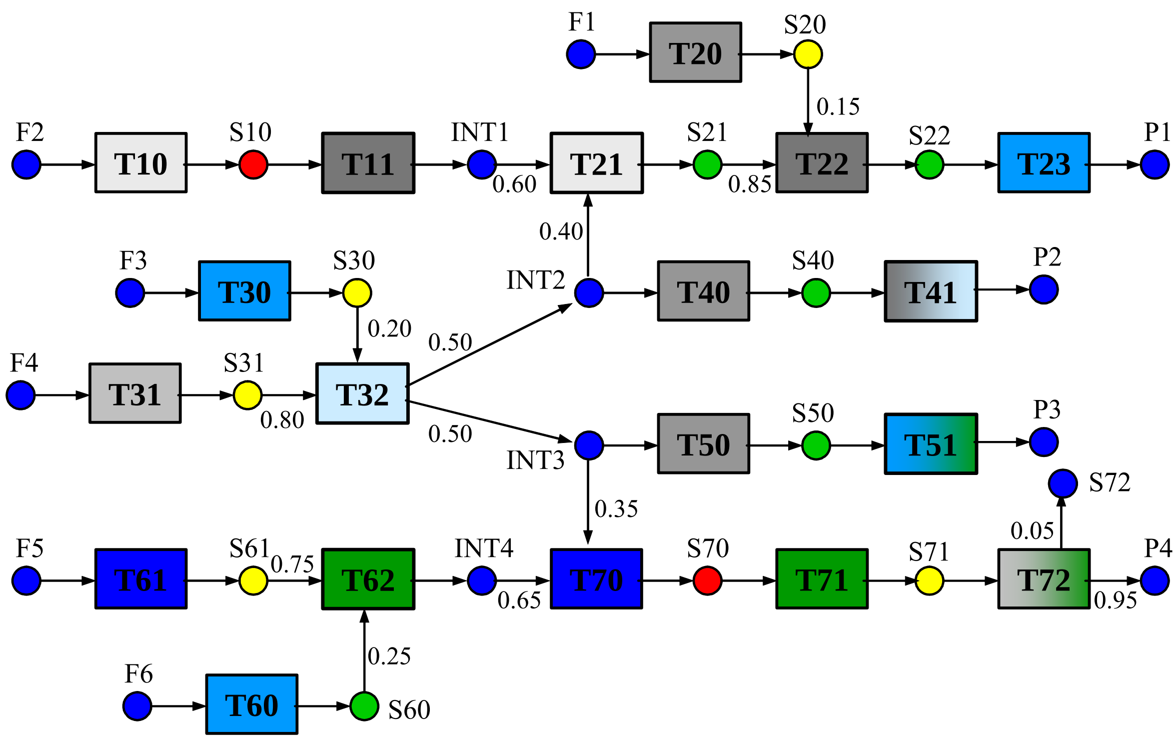


Fig. 11.10. State-Task-Network example.

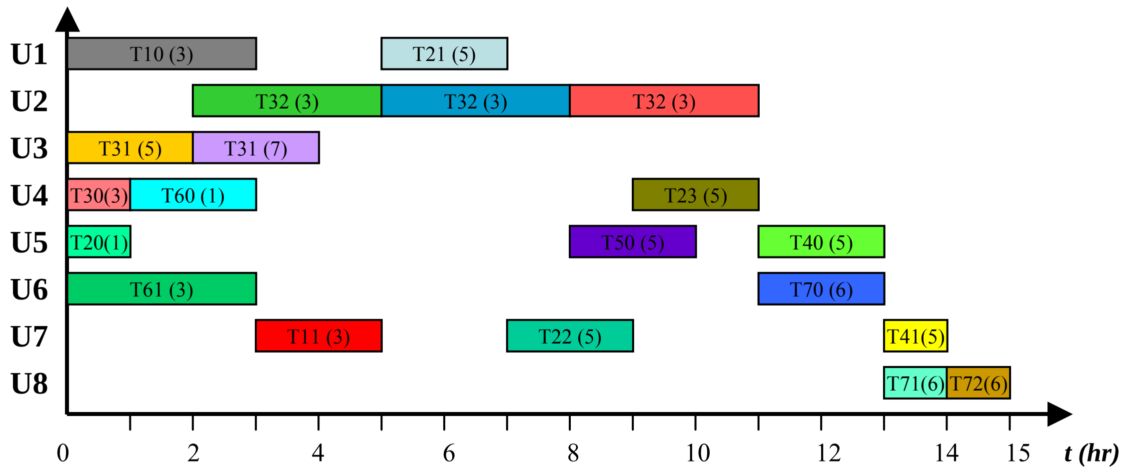


Fig. 11.11. Optimal schedule.