Disjunctive Multiperiod Optimization Methods for Design and Planning of Chemical Process Systems

Susara A. van den Heever[*] and Ignacio E. Grossmann[**]

Department of Chemical Engineering, Carnegie Mellon University

Pittsburgh, PA, 15213

July 1998 / Rev. April 1999

Abstract

In this paper, we present a general disjunctive multiperiod nonlinear optimization model, which incorporates design, as well as operation and expansion planning, and takes into account the corresponding costs incurred in each time period. This model is formulated with the use of logic and disjunctive programming, and includes Boolean variables for design, operation planning and expansion planning. We propose two algorithms for the solution of these problems. The first is a logic-based Outer Approximation (OA) algorithm for multiperiod problems. The second is a bilevel decomposition algorithm, that exploits the problem structure by decomposing it into an upper level design problem and a lower level operation and expansion planning problem, each of which is solved with the logic-based OA algorithm. Applications are considered in the areas of design and planning of process networks, as well as retrofit design for multiproduct batch plants. The results show that the disjunctive logic-based OA algorithm performs best for small problems, while the disjunctive bilevel decomposition algorithm is superior for larger problems. In both cases, a significant decrease in MILP solution time and total solution time is achieved compared to DICOPT++. Results also show that problems with a significant number of time periods can be solved in reasonable time.

[*] E-mail: susara@andrew.cmu.edu

[**] Author to whom correspondence should be addressed. E-mail: grossmann@cmu.edu

1. Introduction

Multiperiod optimization models for design and planning in the chemical industry have received considerable attention in recent years (e.g. Dedopoulos and Shah, 1996; Iyer and Grossmann, 1997; Iyer and Grossmann, 1998; Papalexandri and Pistikopoulos, 1994; Paules and Floudas, 1992; Sahinidis *et al.*, 1989; Sahinidis and Liu, 1996; Varvarezos *et al.*, 1992). Multiperiod plants are process plants where costs and demands typically vary from period to period due to market or seasonal changes (Fig. 1). Examples of multiperiod plants include refineries, utility systems and oil production platforms.

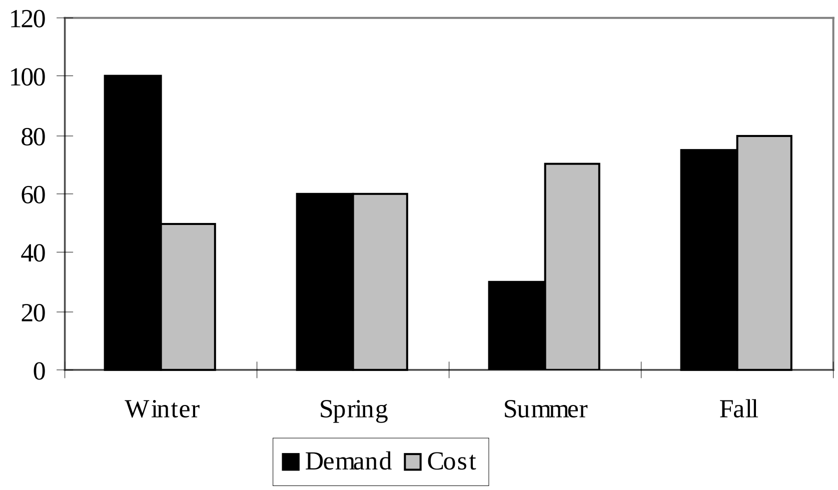

Figure 1: Seasonal changes in demand and cost

Models for multiperiod optimization typically have an objective, such as minimizing cost, subject to constraints in the form of equations and usually involve both continuous and discrete variables. Continuous variables can be either state variables representing operating conditions such as time dependent flows and temperatures, or design variables representing equipment sizes. Discrete variables can be binary (0 or 1) or Boolean (true or false) and represent discrete decisions, for instance to invest in a unit or to operate a unit in a given period. Equations representing these models can be linear, for example mass balances, or nonlinear, for example process performance equations. Constraints can be valid for all periods or for an individual period, and some variables and/or constraints may link the time periods (Fig. 2), preventing a decomposition solution where time periods are solved independently. Models involving 0-1 and continuous variables with nonlinear constraints are classified as Mixed Integer Nonlinear Programming (MINLP) problems. MINLP problems are notorious for being NPcomplete (Garey and Johnson, 1978), meaning they require exponential solution times in the worst case. For multiperiod MINLP models, solution times quickly become intractable, particularly if the models involve an increase in the number of binary variables with each additional period. Thus, there is clearly a need for developing more efficient algorithms and models.

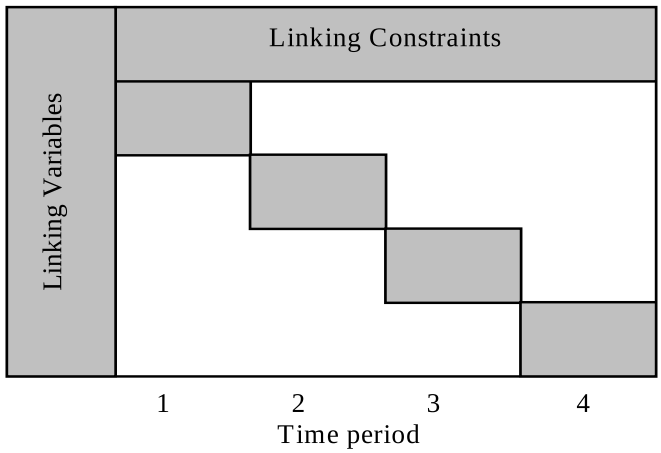

Figure 2: Block-diagonal structure with linking constraints and linking variables

DICOPT++, the MINLP solver used in the commercial optimization software package GAMS (Brooke *et al.*, 1992), is the basis of comparison for the algorithms proposed in this work. It makes use of the outer-approximation (OA) algorithm (Duran and Grossmann, 1986), which involves iterating between a Nonlinear Programming (NLP) subproblem where the binary variables are held fixed and a Mixed Integer Linear Programming (MILP) master problem where the model is linearized at the NLP solution points. A similar decomposition technique for mixed integer problems is the General Benders Decomposition (GBD) (Geoffrion 1972) which involves the solution of NLP sub-problems and pseudo-integer master problems. The latter correspond to the projection of the full mixed-integer space of OA onto the space of binary variables only (Quesada and Grossmann, 1992), and because of this projection, the OA master problem predicts tighter bounds than the GBD master problem. It will be demonstrated in Section 4 that the bottleneck in the computation is often the MILP master problem.

Sahinidis *et al.* (1989) presented a multiperiod MILP model for long range planning in the chemical industry and solved the problem with techniques such as branch and bound, strong cutting planes, GBD and heuristics. Other methods used to solve multiperiod MILPs include constraint generation and projection used in conjunction with a cutting plane algorithm

(Sahinidis and Liu, 1996), and a bilevel decomposition strategy for synthesis and planning in the chemical industry (Iyer and Grossmann, 1998). Dedopoulos and Shah (1996) presented both MILP and MINLP formulations for multiperiod maintenance planning using an aggregation approach to reduce the problem size. Varvarezos *et al.* (1992) proposed an OA based decomposition method for convex multiperiod MINLPs where the NLP subproblem is further decomposed into a linearized master problem and a NLP primal problem with penalties to ensure feasibility. Algorithms based on GBD have also been employed for the solution of multiperiod MINLP synthesis and planning problems (Papalexandri and Pistikopoulos, 1994). Paules and Floudas (1992) proposed a nested solution procedure combining the GBD and OA algorithms and applied it to heat integrated distillation sequences with multiperiod MINLP formulations. None of these methods, however, specifically address the problem of reducing the computational effort involved in solving the MILP master problem.

The above mentioned models have been developed for specific applications. In sections 2 and 3 we present the problem statement and propose a general multiperiod MINLP model for design, operation planning and expansion planning of general chemical process systems. Section 4 contains a motivating example showing the need for more efficient solution methods for multiperiod MINLPs that involve binary variables for each time period. In Section 5 we introduce a disjunctive formulation of the general model and show how this model can be converted into specific models through appropriate manipulations. Disjunctive programming is an alternative representation of mixed integer programming problems and is discussed by Raman and Grossmann (1994) for the linear case. Motivated by the use of generalized disjunctive programming, we propose a disjunctive OA algorithm in Section 6 for solution of the proposed model. The proposed method is an extension of the logic-based OA algorithm for single period MINLPs (Turkay and Grossmann, 1996a). A second algorithm is proposed in Section 7, based on the bilevel decomposition algorithm of Iyer and Grossmann (1998) involving a higher level design problem and a lower level planning problem. While Iyer and Grossmann's method was restricted to MILPs and the subproblems were solved with the commercial solver CPLEX, the proposed disjunctive bilevel decomposition algorithm is applicable to MINLPs and the sub-problems are solved with the disjunctive OA algorithm. The effectiveness of these algorithms is demonstrated in Section 8 with four examples, ranging from a small illustrative example to an industrial size problem, in the areas of process planning and retrofit design of batch plants. Conclusions are discussed in Section 9.

2. Problem Statement

Given is a process network superstructure operating over T time periods. These can correspond to months, seasons, years, or to periods from discretized stochastic problems or from multiple scenarios relating different operating conditions. Demands, costs, feed conditions, and technical parameters vary from period to period. Deterministic data for these parameters are assumed for the model. The problem involves optimizing the process over all time periods and has an objective of minimizing cost or maximizing profit subject to constraints. Constraints include mass balances, heat balances, process performance equations, linking constraints and logical relationships between processes and decision variables. We consider three main decisions in the optimization problem:

a) Selection of a network structure
b) Operation or shutdown of a process in period t
c) Expansion of a process in period t

Decision (a) is a one-time decision involving investment costs, while the other two decisions are valid for every time period and can involve both fixed and variable operation and expansion costs. Constraints are both linear and nonlinear, while variables are discrete (binary decision variables) and continuous (flows, equipment sizes, etc.). Therefore, the model falls under the general class of multiperiod MINLPs. A general model for design and operation/expansion planning is presented next and applied to a specific example. This model will serve as motivation for developing a more efficient model formulation and solution methods.

3. MINLP model

Consider optimizing a given process network superstructure over time periods $t = 1 \ldots T$. A general model for design and operation/expansion planning of such a superstructure is as follows. We first define the sets, parameter and variables for the process network as shown below:

Sets:

I set of streams

J	set of processes
T	set of time periods

Indices:

i	stream in set I
j	process in set J
t	time period in set T

Parameters:

α_{jt}	variable expansion cost for unit j in period t
β_{jt}	fixed expansion cost for unit j in period t
γ_{jt}	fixed operating cost for unit j in period t
c_{it}	cost associated with stream i in period t
U	valid upper bounds for corresponding variables

Variables:

Binary decision variables

y_j	Selection of investment in unit j
w_{jt}	Operation of unit j in period t
z_{jt}	Capacity expansion of unit j in period t

Continuous decision variables

Q_{jt}	Capacity of unit j in period t
QE_{jt}	Capacity expansion of unit j in period t
x_t	State variables in period t
x_{it}	Subset of state variables for stream i in period t
CO_{jt}	Operating cost of unit j in period t
CE_{jt}	Expansion cost of unit j in period t

Based on the above definitions, the model is as follows:

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it} \qquad (1)$$

subject to

$$
\begin{array}{lll}
CO_{jt} = \gamma_{jt} w_{jt} & \forall j,t & (2) \\
CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} & \forall j,t & (3) \\
g_t(x_t, x_{t-1}) \le 0 & \forall t & (4) \\
h_{jt}(Q_{jt}, x_t, x_{t-1}) \le 0 & \forall j,t & (5) \\
Q_{jt} = Q_{j,t-1} + QE_{jt} & \forall j,t & (6) \\
x_t - Uw_{jt} \le 0 & \forall j,t & (7) \\
QE_{jt} - Uz_{jt} \le 0 & \forall j,t & (8) \\
y_j \le \sum_t w_{jt} & \forall j & (9a) \\
w_{jt} \le y_j & \forall j,t & (9b) \\
w_{jt} \le \sum_{\tau=1}^{t} z_{j\tau} & \forall j,t & (9c) \\
z_{jt} \le w_{jt} & \forall j,t & (9d) \\
Ey \le e & & (10) \\
CO, CE, Q, QE, x \ge 0 & y, w, z \in \{0,1\} & \\
j \in J \qquad i \in I & t \in T &
\end{array}
\qquad (P)
$$

Equation (1) is the objective function minimizing total cost, while (2) and (3) represent operating and expansion costs, respectively. Global constraints valid for a particular period, such as mass balances over mixers, are represented by (4). Constraints represented by (5) are valid for a given unit *j* in a particular period, for example unit input-output relationships. Note that both (4) and (5) may generally involve "pass-on" variables, x_{t-1}, from a previous time period, in which case they will give rise to linking constraints. Only a small subset of equations involve "pass-on" variables, if they are present at all, and hence the block-diagonal structure of Fig. 2. Additional linking constraints are represented by (6) and state that the capacity at the current period equals the capacity at the previous period plus the capacity expansion. Equation (7) sets all state variables associated with unit *j* to zero if it is not operated in period *t*, while equation (8) sets capacity expansion of unit *j* to zero if it is not expanded in period *t*. Equations (9a)-(9d) represent logical relationships between the binary variables. Equation (9a) states that a unit is operated in at least one period if it is selected, while (9b) states that a unit must be selected if it is operated in any period. Equation (9c) states that a unit can only be operated if it is already expanded beyond zero capacity, and (9d) states that a unit may only be expanded in a certain period if it is operated. The assumption here is that unit *j* is only expanded in period *t* if it needs to be operated in period *t*. This is a realistic assumption, seeing that there will be no need for expansion if unit *j* is never operated. The construction work for an expansion has to be completed before operation starts, and is dealt with by discounting the expansion costs appropriately. Equation (10) represents logic propositions relating binary design variables, *y*, for the topology of the network (combinations of units that are permitted). In case of no expansion (z_{jt}=0), (1) becomes min $\sum_t \sum_j CO_{jt} + \sum_j CI_j + \sum_t \sum_i c_{it} x_{it}$ and (5) becomes $CI_j = \beta_j y_j + \alpha_j Q_j$ where $Q_j \geq Q_{jt}$ for all *t*. In all problem formulations hereafter, the domains $j \in J$, $i \in I$ and $t \in T$ apply, but are omitted for the sake of convenience.

It should be noted that while the binary variables y_j are related to the selection of a unit independent of time period, the binary variables w_{jt} and z_{jt} for operation and expansion are defined for each time period. Hence the number of these variables will increase as the number of time periods increase.

4. Example 1

Consider optimizing the process network superstructure given in Figure 3 over time periods t = 1..21. Demands and costs differ from period to period with an increasing trend, and there exists a trade-off between costs and process efficiency. It is required to find an optimal structure in which only one of processes 2 and 3 are selected, or neither, in which case raw material B is purchased. It is also required to find the optimal operational and capacity expansion plan. Due to the large amount of data for the examples, they are not presented in this paper. Readers interested in the data can contact the authors.

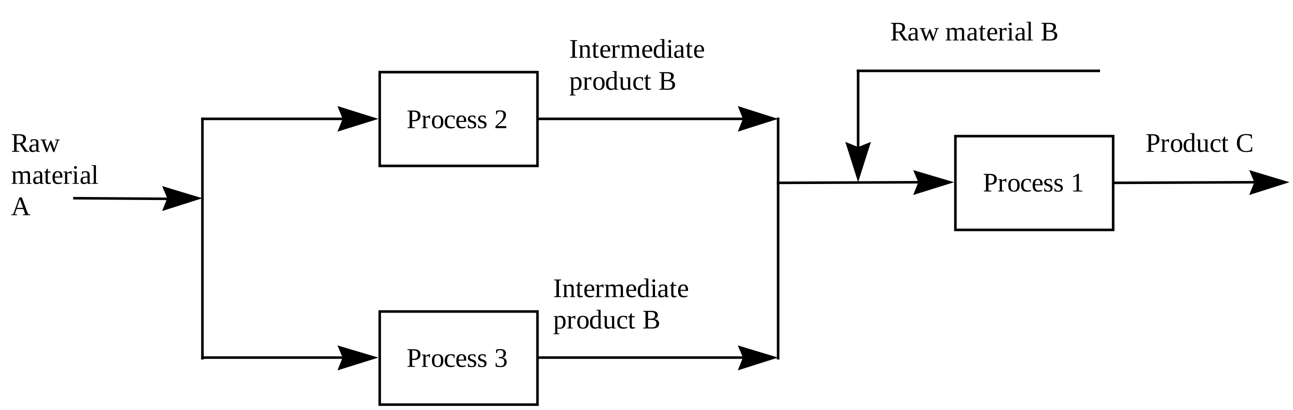

Figure 3: Process network superstructure (Kocis and Grossmann, 1989)

Figure 4 shows the optimal solution for the above example over five time periods. Other specific solutions are not discussed, since the intention of this example is to illustrate the computational difficulties.

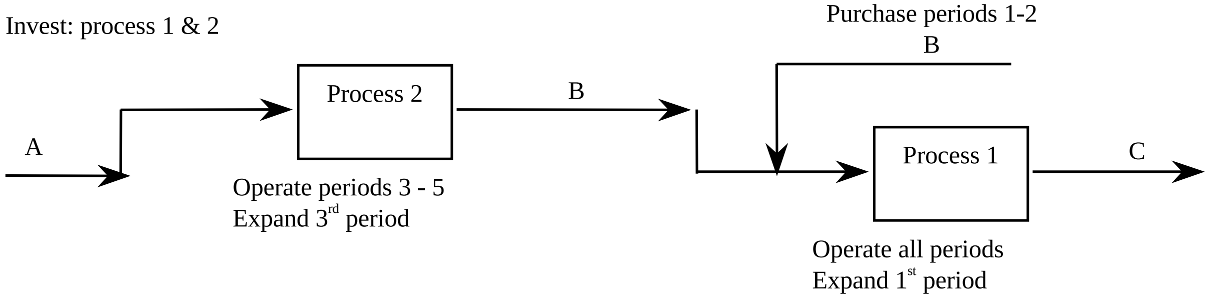

Figure 4: Example of solution over 5 time periods

In applying model (P) to this superstructure, and using DICOPT++ to obtain the solution, it becomes apparent that the major bottleneck is the solution of the MILP master problem. This is due to the increase in binary variables with each additional time period, as shown in Table 1. The NLP subproblems do not contain any binaries and are therefore not as difficult to solve as the MILP master problem. As seen in Figure 5, the MILP solution time increases nearly exponentially versus the modest increase of the NLP time. Note the maximum of 9.7 CPU seconds for solving the NLP subproblem versus the near 500 seconds required for solving the MILP master problem at 21 time periods. The rest of the paper concentrates on addressing the problem of reducing the computational time of the master problem.

Table 1: Growth in problem size

Time Periods	Discrete Variables	Continuous Variables	Constraints
1	9	18	38
5	33	74	154
9	57	130	270
13	81	186	386
17	105	242	502
21	129	298	618

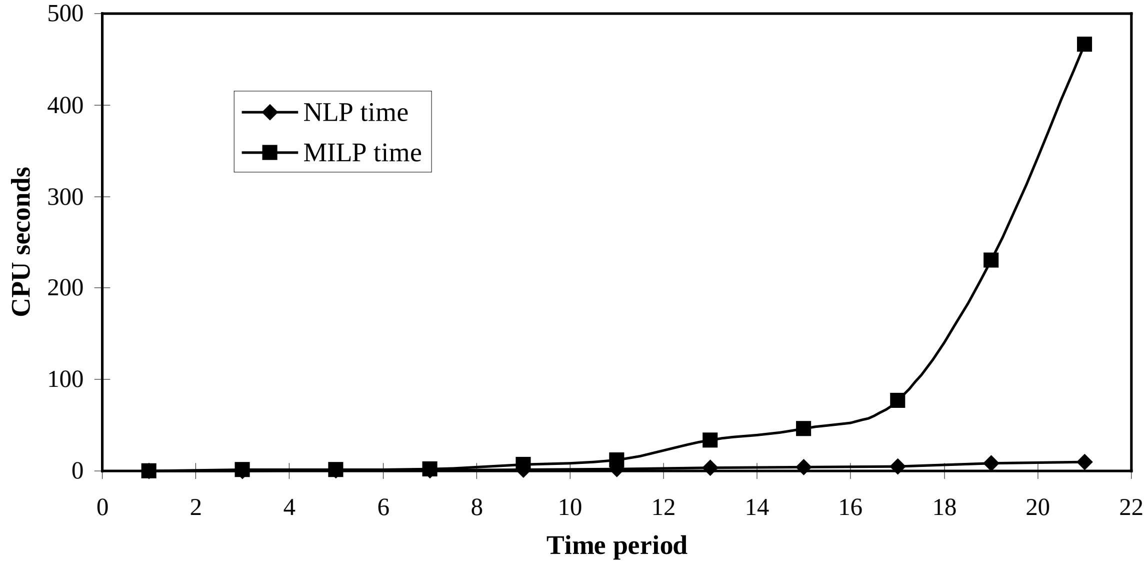

Figure 5: Solution time for the MILP master problem vs. the NLP subproblem

5. Disjunctive Multiperiod Model

5.1 Background

Raman and Grossmann (1994) demonstrated the representation of mixed-integer logic through generalized disjunctive programming. Consider the disjunction (a set of constraints of which at least one must be valid) represented here with the use of the logic operators OR (∨) and NOT (¬):

$$\begin{bmatrix} w_{jt} \\ h_{jt}(x) \le 0 \\ c_{jt} = \alpha_{jt} \end{bmatrix} \vee \begin{bmatrix} \neg w_{jt} \\ B^{jt} x = 0 \\ c_{jt} = 0 \end{bmatrix} \qquad \forall j,t$$

The above representation can be interpreted as follows: If unit *j* is operated in period *t* (w_{jt} = true), then enforce equation $h_{jt}(x)$ describing that unit and apply a fixed cost α_{jt}. If the unit is not operated (w_{jt} = false), the fixed cost and a subset of continuous variables are set to zero through the matrix B^{jt}. The logical relationships among discrete variables describing connections and interactions between units are given through logic propositions consisting of the variables and the logic operators (such as OR (∨), AND (∧), NOT (¬) and IMPLY (⇒)). For example,

$$y_1 \Rightarrow y_2 \wedge y_3$$

means that selection of unit 1 implies selection of unit 2 and selection of unit 3. Likewise,

$$y_j \Rightarrow \vee_t z_{jt}$$

means that selection of unit *j* implies expansion of unit *j* in at least one of the periods.

5.2 Model

In the following derivation, we convert model (P) step by step to its disjunctive form. First consider only the expansion planning and associated equations and variables:

$$\begin{bmatrix} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{bmatrix} \vee \begin{bmatrix} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{bmatrix} \qquad \forall j,t \qquad \text{(C1)}$$

(C1) implies that if unit *j* is expanded in period *t*, equations (3) and (6) are applied and if not, the capacity simply equals the capacity from the previous period and no cost is incurred. Next we consider the higher level decision of operation planning:

$$\begin{bmatrix} w_{jt} \\ h_{jt}(Q_{jt}, x_t, x_{t-1}) \le d \\ CO_{jt} = \gamma_{jt} \\ Expansion \quad equations \quad (C1) \end{bmatrix} \vee \begin{bmatrix} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{bmatrix} \qquad \forall j,t \qquad \text{(C2)}$$

(C2) implies that if unit *j* is operated in period *t*, the appropriate equations (2) and (5) are applied, and if not, a subset of variables associated with that unit and operation costs are set to zero. Note that disjunctions (C1) are embedded in (C2), seeing that an expansion will only be considered if the unit is operated in the given time period (w_{jt} = True). The construction of the expansion is started some time before operation starts, and is dealt with through appropriate discounting of expansion costs. Finally, we add the design level of decision making:

$$\begin{bmatrix} y_j \\ Operation \quad equations \quad (C2) \\ and \\ (Expansion \quad equations \quad (C1) \\ or \\ CI_j = \beta_j + \alpha_j Q_j) \end{bmatrix} \vee \begin{bmatrix} \neg y_j \\ B^{jt} x_t = 0 \\ CI_j = 0 \\ \forall t \end{bmatrix} \quad \forall j \qquad \text{(C3)}$$

(C3) implies that if it is decided to install unit *j*, (y_j = True), the appropriate equations are applied, and if not, a subset of variables associated with that unit are set to zero for all periods and no cost is incurred. Disjunctions (C2) and (C1) are embedded in (C3), seeing that an operational and expansion plan is only necessary if it is decided to invest in a process. Furthermore, if the expansion is not to be included in the problem, (C1) is simply replaced by the investment cost function. By substituting the applicable equations into (C3), adding all remaining equations from (P), and expressing constraints (9a)-(10) in their propositional logic form (see Raman and Grossmann, 1994) we obtain the final disjunctive formulation of the multiperiod model (P).

Model (PD):

$$\min \quad Z = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t(x_t, x_{t-1}) \le 0 \qquad \forall t$$

$$\left[\begin{array}{c} y_j \\ \left\{ \left[\begin{array}{c} w_{jt} \\ h_{jt}(Q_{jt}, x_t, x_{t-1}) \le d \\ CO_{jt} = \gamma_{jt} \\ \left[\begin{array}{c} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array} \right] \vee \left[\begin{array}{c} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array} \right] \end{array} \right] \vee \left[\begin{array}{c} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array} \right] \right\} \forall t \end{array} \right] \vee \left[\begin{array}{c} \neg y_j \\ B^{jt} x_t = 0 \\ \forall t \end{array} \right] \qquad \forall j$$

$$y_j \Rightarrow \bigvee_{t=1}^{T} w_{jt} \quad \forall j,t, \qquad w_{jt} \Rightarrow y_j \quad \forall j,t$$

$$w_{jt} \Rightarrow \bigvee_{\tau=1}^{t} z_{j\tau} \quad \forall j,t, \qquad z_{jt} \Rightarrow w_{jt} \quad \forall j,t$$

$$\Omega(y) = True$$

$$CE, CO, Q, QE, x \ge 0 \qquad y, w, z \in \{True, False\}$$

Model (PD) is a general representation incorporating three levels of decision making, namely design, operation planning and expansion planning. These decisions are represented by the discrete variables, y_j, w_{jt} and z_{jt}, respectively and the formulation can easily be converted to a more specific model by ignoring the appropriate terms and variables. This is illustrated for three particular classes of problems where these formulations apply:

i) Multiperiod design (only *y*). This problem involves only discrete decisions y_j for the selection of the topology of a process network (e.g. Papalexandri and Pistikopoulos, 1994):

$$\min \quad Z = \sum_j CI_j + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t(x_t, x_{t-1}) \le 0 \qquad \forall t$$

$$\begin{bmatrix} y_j \\ h_{jt}(Q_{jt}, x_t, x_{t-1}) \le d \\ Q_j \ge Q_{jt} \\ CI_j = \beta_j + \alpha_j Q_j \end{bmatrix} \vee \begin{bmatrix} \neg y_j \\ B^{jt} x_t = 0 \\ CI_j = 0 \\ \forall t \end{bmatrix} \quad \forall j \qquad \text{(P1)}$$

$$\Omega(y) = True$$

$$CI, Q, x \ge 0, \quad y \in \{True, False\}$$

ii) Multiperiod design and capacity planning (y and z). This problem involves the selection of the topology (y_j) of a process network, as well as the potential capacity expansion at each time period (z_{jt}) (e.g. Iyer, 1997; Sahinidis *et al.*, 1989; Varvarezos *et al.*, 1992):

$$\min \quad Z = \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t(x_t, x_{t-1}) \le 0 \qquad \forall t$$

$$\begin{bmatrix} y_j \\ h_{jt}(Q_{jt}, x_t, x_{t-1}) \le d \\ \begin{bmatrix} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{bmatrix} \vee \begin{bmatrix} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{bmatrix} \forall t \end{bmatrix} \vee \begin{bmatrix} \neg y_j \\ B^{jt} x_t = 0 \\ \forall t \end{bmatrix} \quad \forall j$$

$$y_j \Rightarrow \bigvee_{t=1}^{T} z_{jt} \quad \forall j, \qquad z_{jt} \Rightarrow y_j \quad \forall j, t \qquad \text{(P2)}$$

$$\Omega(y) = True$$

$$CE, Q, QE, x \ge 0, \quad y, z \in \{True, False\}$$

iii) Multiperiod Planning (only w). This problem arises in process networks with fixed topology and fixed capacities in which decisions involve the start-up/shutdown of processes (w_{jt}) (e.g. Iyer and Grossmann, 1997):

$$\min \quad Z = \sum_t \sum_j CO_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t(x_t, x_{t-1}) \leq 0 \qquad \forall t$$

$$\begin{bmatrix} w_{jt} \\ h_{jt}(Q_{jt}, x_t, x_{t-1}) \leq d \\ CO_{jt} = \gamma_{jt} \end{bmatrix} \vee \begin{bmatrix} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{bmatrix} \qquad \forall j, t \qquad \text{(P3)}$$

$$\Omega(w) = True$$

$$CO, Q, x \geq 0, \quad w \in \{True, False\}$$

6. Disjunctive OA Algorithm

Turkay and Grossmann (1996a) proposed a logic-based OA algorithm based on the OA method by Duran and Grossmann (1986) which involves iteration between an NLP subproblem where all binary variables are fixed, and an MILP master problem where the nonlinear equations are relaxed and linearized at the NLP solution points. In the logic-based algorithm, an initial set covering problem is solved to determine the least number of possible configurations, *N*, to cover all units. These *N* configurations are used to solve *N* initial NLP subproblems to generate initial linearizations for all equations. Alternatively, the relaxed MINLP can be used as starting point if the set covering is not applicable, for example in the case of the retrofit design problem where an initial configuration is given. The NLP subproblems only include equations for existing units (i.e. equations of disjunctions with true value). The algorithm that is proposed in this section (see Fig. 6) is an extension to multiperiod problems of Turkay and Grossmann's method, which was restricted to a single period.

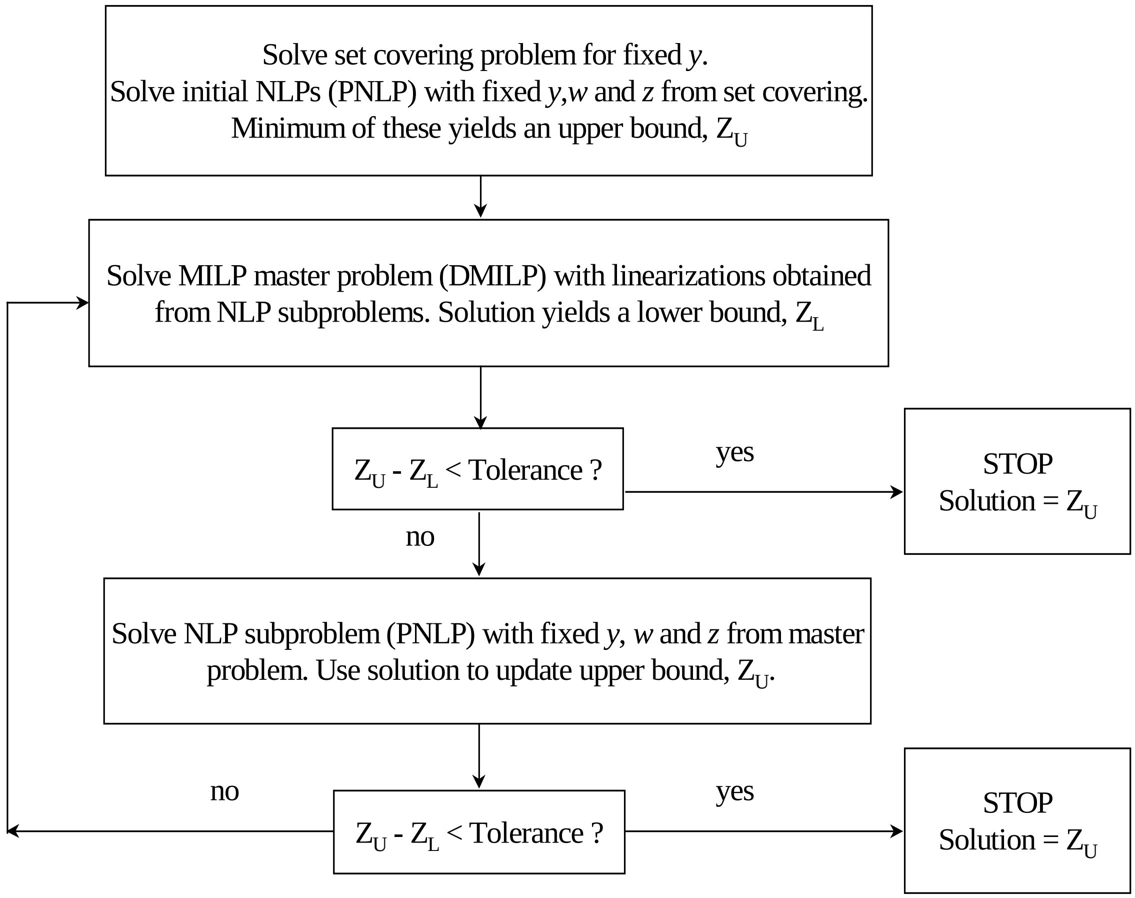

Figure 6: Disjunctive logic-based OA algorithm as applied in this work

In applying this algorithm, we decompose model (PD) into an NLP sub-problem and MILP master problem, as shown below. Both problems are converted to mixed-integer form through the convex hull formulation (Balas, 1985; see also Turkay and Grossmann, 1996a). The advantage of this formulation is that it reduces the dimensionality of the problem by only considering disjunctions for which the Boolean variable is true (see (PNLP) below), avoids singularities due to linearizations at zero flows, and eliminates non-convexities of non-existing processes.

The NLP sub-problem is the following:

$$\min \quad Z_U = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$
\begin{array}{l}
g_t(x_t, x_{t-1}) \le 0 \\
\left.\begin{array}{l}
\left.\begin{array}{l}
\left.\begin{array}{l} Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array}\right\} z_{jt} = True \\
\left.\begin{array}{l} Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array}\right\} z_{jt} = False \\
h_{jt}(Q_{jt}, x_t, x_{t-1}) \le d \\
CO_{jt} = \gamma_{jt}
\end{array}\right\} w_{jt} = True \\
\left.\begin{array}{l} B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array}\right\} w_{jt} = False
\end{array}\right\} y_j = True \\
\left. B^{jt} x_t = 0 \right\} y_j = False \\
CE, CQ, QE, Q, x \ge 0, \qquad y, w, z \in \{True, False\}
\end{array} \qquad \text{(PNLP)}
$$

To formulate the MILP master problem, x_t is partitioned into two sets of variables, namely $x_{t,z}$, which is driven to zero if the Boolean associated with the disjunction is false, and $x_{t,nz}$, which can take on any positive value irrespective of the Boolean value. Furthermore, the equations $h_{jt}(Q_{jt}, x_t, x_{t-1})$ and $g_t(x_t, x_{t-1})$ are partitioned into a set of nonlinear equations, $h_{jt}^{nl}(x_t) \le 0$ and $g_t^{nl}(x_t) \le 0$, and a set of linear equations $h_{jt}^{l}(Q_{jt}, x_t, x_{t-1}) = Ax_{t,nz} + Bx_{t,z} + CQ_{jt} \le b$ and $g_t^{l}(x_t) = Dx_t \le d$. For simplicity in the presentation, the pass-on variables, x_{t-1}, are not considered, although it can easily be shown that the derivation would be the same if the variables x_{t-1} were present. Given L major iterations, the linearization set for active terms of the disjunctions for operation is defined as $K^{jt} = \{k \mid w_{jt} = True, k = 1..L, t = 1..T\}$. The formulation of the disjunctive master problem is as follows (DLP).

$$
\min \quad Z_L = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}
$$

subject to

$$g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \le 0 \quad \forall t$$
$$Dx_t \le d \qquad \forall t$$

$$\left[\begin{array}{c} y_j \\ \left[\begin{array}{c} w_{jt} \\ h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T (x_{t,nz} - x_{t,nz}^k) + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T (x_{t,z} - x_{t,z}^k) \le a \\ Ax_{t,nz} + Bx_{t,z} + CQ_{jt} \le b \\ k \in K^j \\ CO_{jt} = \gamma_{jt} \\ \left[\begin{array}{c} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array}\right] \vee \left[\begin{array}{c} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array}\right] \end{array}\right] \vee \left[\begin{array}{c} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array}\right] \forall t \end{array}\right] \vee \left[\begin{array}{c} \neg y_j \\ B^{jt} x_t = 0 \\ \forall t \end{array}\right] \quad \forall j$$

$$y_j \Rightarrow \mathop{\vee}_{t=1}^{T} w_{jt} \quad \forall j,t, \qquad\qquad w_{jt} \Rightarrow y_j \qquad \forall j,t$$
$$w_{jt} \Rightarrow \mathop{\vee}_{\tau=1}^{t} z_{j\tau} \quad \forall j,t, \qquad\qquad z_{jt} \Rightarrow w_{jt} \qquad \forall j,t$$
$$\Omega(y) = True$$
$$CE, CO, Q, QE, x \ge 0, \qquad y, w, z \in \{True, False\}$$

After applying the convex hull formulation to each disjunction, the final formulation is obtained (DMILP) (see Appendix A for the full derivation).

$$\min \qquad Z_L = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$\nabla_{x_t} g_t^{nl}(x_t^k)^T x_t \le -g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T x_t^k \qquad \forall t$$

$$Dx_t \le d \qquad \forall t$$

$$\nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^1 + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z} \le -\left[h_{jt}^{nl}(x_t^k) - \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^k - \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z}^k - a\right] w_{jt} \qquad \forall j,t$$

$$k \in K^j$$

$$Ax_{t,nz}^1 + Bx_{t,z} + CQ_{jt}^1 \le bw_{jt} \qquad \forall j,t$$

$$x_{t,nz} = x_{t,nz}^1 + x_{t,nz}^2 \qquad \forall t$$

$$x_{t,nz}^1 \le Uw_{jt} \qquad \forall j,t$$

$$x_{t,nz}^2 \le U(1-w_{jt}) \qquad \forall j,t$$

$$x_{t,z} \le Uw_{jt} \qquad \forall j,t$$

$$Q_{jt} = Q_{jt}^1 + Q_{jt}^2 \qquad \forall j,t$$

$$Q_{jt}^1 \le Uw_{jt} \qquad \forall j,t$$

$$Q_{jt}^2 \le U(1-w_{jt}) \qquad \forall j,t$$

$$CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} \qquad \forall j,t$$

$$CO_{jt} = \gamma_{jt} w_{jt} \qquad \forall j,t$$

$$Q_{jt} = Q_{j,t-1} + QE_{jt} \qquad \forall j,t$$

$$QE_{jt} - Uz_{jt} \le 0 \qquad \forall j,t$$

$$y_j \le \sum_{t=1}^{T} w_{jt} \qquad \forall j$$

$$w_{jt} \le y_j \qquad \forall j,t$$

$$w_{jt} \le \sum_{\tau=1}^{t} z_{j\tau} \qquad \forall j,t$$

$$z_{jt} \le w_{jt} \qquad \forall j,t$$

$$Ey \le e$$

$$CE, CO, Q, QE, x \ge 0, \qquad y, x, z \in \{0,1\}$$

$$CE, CO, Q, QE, x \in \Re$$

Remarks:

- For the initial NLPs the operation (w_{jt}) and expansion (z_{jt}) Boolean variables are set to true for all periods if a unit is selected from the set-covering problem. For all other NLPs the Boolean variables are fixed at values from the MILP master problem (DMILP).
- Branching priorities can be used for the branch and bound search in the master problem, exploiting the hierarchical structure of the problem by branching first on y_j, then on w_{jt} and then on z_{jt}.
- Convergence to the optimal solution is guaranteed in a finite number of iterations if the problem is convex.

7. Bilevel Decomposition Algorithm

In this section, we consider the further decomposition of problem (PD) in an effort to improve on the algorithm presented in the previous section. Similarly to Iyer and Grossmann (1998), the design and planning problem is decomposed into an aggregated upper level design problem (DP) that includes only the binary variables y_j, and a lower level operation and expansion planning problem (OEP) which corresponds to problem (PD) for fixed value of y_j. The solution to (DP) yields a lower bound and a fixed configuration to be used in (OEP). An upper bound is obtained from (OEP). Previous values of *y*, *QE* and *x* are used to formulate cuts for problem (DP) (see Iyer and Grossmann (1998)). Integer cuts in *y* are used to exclude subsets and supersets of previously obtained feasible configurations and to exclude infeasible configurations from future calculations. Design cuts in *QE*, *x* and *y* are used to force values of state variables in (DP) to be greater than or equal to their values in (OEP) if the same configuration is chosen in both problems. The solution of (OEP) with the current upper bound is the final solution after convergence is achieved. The design problem (DP) is given by:

$$\min \quad Z^{DP} = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t(x_t) \le 0 \qquad \forall t$$

$$\begin{bmatrix} y_j \\ h_{jt}(Q_{jt}, x_t) \le 0 \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CO_{jt} \ge \gamma_{jt} \dfrac{x_t}{x_{t,upper}} \\ CE_{jt} \ge \alpha_{jt} QE_{jt} + \beta_{jt} \dfrac{QE_{jt}}{QE_{jt,upper}} \\ \forall t \end{bmatrix} \vee \begin{bmatrix} \neg y_j \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \\ CE_{jt} = 0 \\ Q_{jt} = 0 \\ \forall t \end{bmatrix} \qquad \forall j \qquad \text{(DP)}$$

$$\Omega(y) = True, \qquad CE, CO, QE, Q, x \ge 0$$
$$CE, CO, QE, Q, x \in \Re, \qquad y \in \{True, False\}$$

(DP) is a relaxation of the original problem, since w_{jt} and z_{jt} are relaxed as described in Appendix B and shown above. This ensures that the optimal objective value Z^{DP} corresponds to a lower bound. (OEP) is in a reduced solution space, since a subset of fixed *y* variables are used as obtained from (DP), making it possible to ignore a large number of equations and variables.

$$\min \quad Z^{OEP} = \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t(x_t) \le 0 \qquad \forall t$$

$$\begin{bmatrix} w_{jt} \\ h_{jt}(Q_{jt}, x_t) \le 0 \\ CO_{jt} = \gamma_{jt} \\ \begin{bmatrix} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{bmatrix} \vee \begin{bmatrix} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{bmatrix} \end{bmatrix} \vee \begin{bmatrix} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{bmatrix} \qquad \forall j \in J^1, t$$

$$w_{jt} \Rightarrow \bigvee_{\tau=1}^{t} z_{j\tau} \quad \forall j,t \qquad\qquad z_{jt} \Rightarrow w_{jt} \quad \forall j,t$$

$$\neg y_{j,fixed} \Rightarrow \neg w_{jt} \quad \forall j,t \qquad\qquad \Omega(w) = True \qquad\qquad \text{(OEP)}$$

$$CE, CO, QE, Q, x \ge 0$$

$$CE, CO, QE, Q, x \in \Re$$

$$w, z \in \{True, False\}$$

where $J^1 = \{j |\, y_j = 1\}$.

8. Disjunctive Bilevel Decomposition Algorithm

Iyer and Grossmann's work was restricted to linear problems and the sub-problems were formulated in the full space as mixed integer problems and solved with a branch and bound method for MILP. The method proposed here (see Fig. 7) is applicable to nonlinear problems and are solved with the disjunctive logic-based OA algorithm for which MILP master problems (MIPDP) and (MIPOEP) can be derived by applying the convex hull to each disjunction (see Appendix B). In this work, the operation and expansion planning are incorporated into one sub-problem, (OEP), whereas Iyer and Grossmann considered these planning decisions in different models. The basic idea is that an outer loop iterates between (DP) and (OEP), similar to the algorithm of Iyer and Grossmann, while both (DP) and (OEP) are solved through inner loops using the disjunctive algorithm presented in Section 6. For (DP), the initial set covering problem is solved only once in the first iteration and after that linearizations and cuts are added directly to the MILP master problem. For (OEP) no initial set covering is needed, since a subset of *y* variables is considered as obtained from (DP).

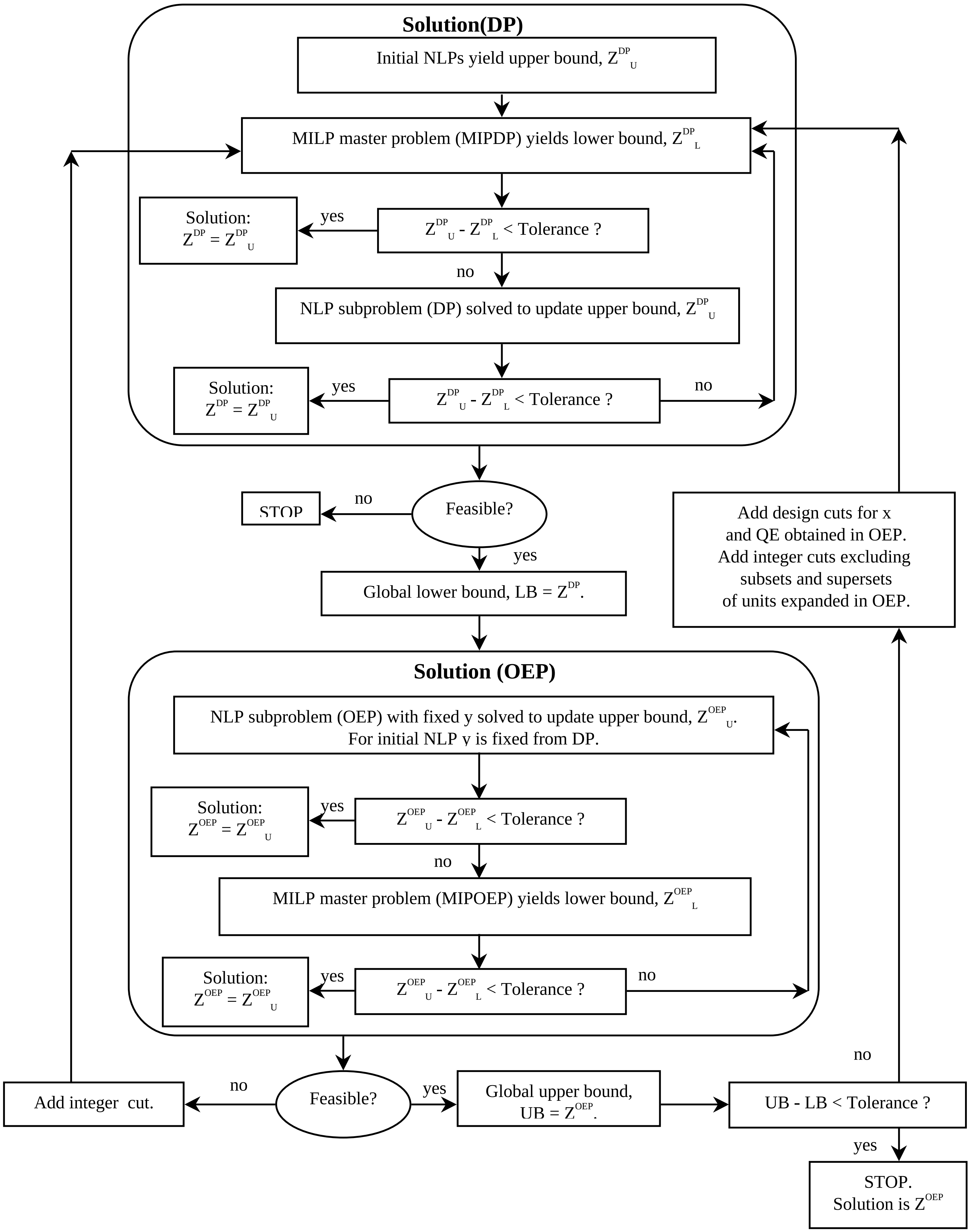

Figure 7: Disjunctive bilevel decomposition algorithm

Remarks:

- An optimal solution is guaranteed in a finite number of iterations if the problem is convex.
- Design cuts in x are always valid if the coefficient in the objective function is positive. Otherwise, the cut might not be valid (Iyer and Grossmann, 1998).
- Integer cuts to exclude subsets and supersets of feasible solutions are formulated with $y_j = 1$ if unit j was expanded in (OEP) and $y_j = 0$ otherwise.

9. Examples

9.1 Example 1 revisited

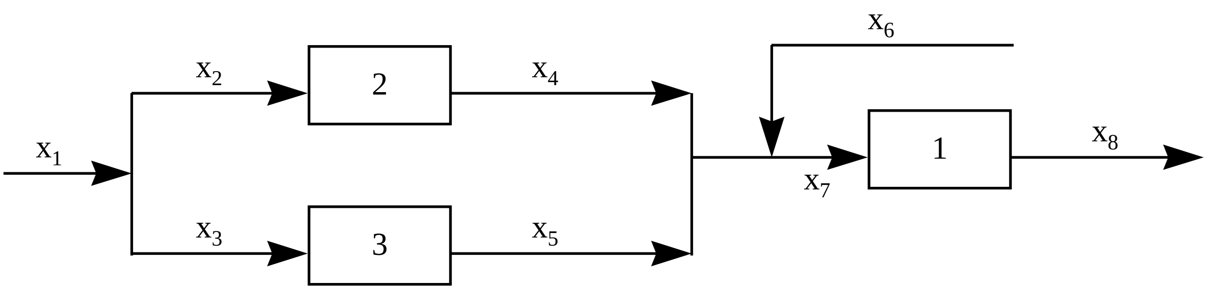

Figure 8: Three unit superstructure

The optimization problem that was considered earlier in the paper can be expressed in the disjunctive modeling framework (PD) as follows:

i) Objective function:

$$\min \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

ii) Mass balances at mixing/splitting points:

$$x_{1,t} - x_{2,t} - x_{3,t} = 0$$
$$x_{4,t} + x_{5,t} + x_{6,t} - x_{7,t} = 0 \qquad \forall t$$

iii) Specifications on the flows and production:

$$x_{t,5} \le 5 \qquad \forall t$$
$$x_{t,8} \le 1 \qquad \forall t$$

iv) Disjunctions for each process:

Process 1:

$$\begin{bmatrix} y_1 \\ \begin{bmatrix} \begin{bmatrix} w_{1t} \\ x_{8,t} = 0.9x_{7,t} \\ CO_{1t} = \gamma_{1t} \\ \begin{bmatrix} z_{1t} \\ Q_{1t} = Q_{1,t-1} + QE_{1t} \\ CE_{1t} = \alpha_{1t} QE_{1t} + \beta_{1t} \end{bmatrix} \vee \begin{bmatrix} \neg z_{1t} \\ Q_{1t} = Q_{1,t-1} \\ CE_{1t} = 0 \end{bmatrix} \end{bmatrix} \vee \begin{bmatrix} \neg w_{1t} \\ x_{7,t} = x_{8,t} = 0 \\ CO_{1t} = 0 \end{bmatrix} \end{bmatrix} \forall t \end{bmatrix} \vee \begin{bmatrix} \neg y_1 \\ x_{7,t} = x_{8,t} = 0 \\ \forall t \end{bmatrix}$$

Process 2:

$$\begin{bmatrix} y_2 \\ \begin{bmatrix} \begin{bmatrix} w_{2t} \\ x_{4,t} = \ln(1 + x_{2,t}) \\ CO_{2t} = \gamma_{2t} \\ \begin{bmatrix} z_{2t} \\ Q_{2t} = Q_{2,t-1} + QE_{2t} \\ CE_{2} = \alpha_{2t} QE_{2t} + \beta_{2t} \end{bmatrix} \vee \begin{bmatrix} \neg z_{2t} \\ Q_{2t} = Q_{2,t-1} \\ CE_{2t} = 0 \end{bmatrix} \end{bmatrix} \vee \begin{bmatrix} \neg w_{2t} \\ x_{2,t} = x_{4,t} = 0 \\ CO_{2t} = 0 \end{bmatrix} \end{bmatrix} \forall t \end{bmatrix} \vee \begin{bmatrix} \neg y_2 \\ x_{2,t} = x_{4,t} = 0 \\ \forall t \end{bmatrix}$$

Process 3:

$$\begin{bmatrix} y_3 \\ \begin{bmatrix} \begin{bmatrix} w_{3t} \\ x_{5,t} = 1.2\ln(1 + x_{3,t}) \\ CO_{3t} = \gamma_{3t} \\ \begin{bmatrix} z_{3t} \\ Q_{3t} = Q_{3,t-11} + QE_{3t} \\ CE_{3t} = \alpha_{3t} QE_{3t} + \beta_{3t} \end{bmatrix} \vee \begin{bmatrix} \neg z_{3t} \\ Q_{3t} = Q_{3,t-1} \\ CE_{3t} = 0 \end{bmatrix} \end{bmatrix} \vee \begin{bmatrix} \neg w_{3t} \\ x_{3,t} = x_{5,t} = 0 \\ CO_{3t} = 0 \end{bmatrix} \end{bmatrix} \forall t \end{bmatrix} \vee \begin{bmatrix} \neg y_3 \\ x_{3,t} = x_{5,t} = 0 \\ \forall t \end{bmatrix}$$

v) Propositional logic and specifications:

$$y_j \Rightarrow \underset{t=1}{\overset{T}{\vee}} w_{jt} \quad \forall j \qquad\qquad w_{jt} \Rightarrow y_j \quad \forall j,t$$

$$w_{jt} \Rightarrow \underset{\tau=1}{\overset{t}{\vee}} z_{j\tau} \quad \forall j,t \qquad\qquad z_{jt} \Rightarrow w_{jt} \quad \forall j,t$$

$$y_2 \Rightarrow y_1, \qquad y_3 \Rightarrow y_1, \qquad \neg y_2 \vee \neg y_3$$

vi) Variables:

$$x_{it}, Q_{jt}, QE_{jt}, CE_{jt}, CO_{jt} \geq 0 \qquad\qquad y_j, w_{jt}, z_{jt} = \{True, False\}$$

$$i = 1,..8 \qquad\qquad j = 1,..3 \qquad\qquad t = 1,..21$$

Formulated as the above generalized disjunctive program, the problem is first solved with the disjunctive logic-based OA algorithm outlined in Figure 6. It is then decomposed into (DP) and (OEP) and solved with the disjunctive bilevel decomposition algorithm outlined in Figure 7. For comparison, the problem is formulated as the MINLP model (P) and solved with DICOPT++. All three methods obtain the same objective value for problems with up to 20 time periods. It is clear from Figure 9 and Table 2 that both the proposed algorithms show significantly improved MILP solution times. This is partly because of the disjunctive formulation where equations and variables are only included if the binary variable for that disjunction equals 1. In addition, the bilevel decomposition leads to two subproblems that are easier to solve than the original problem (PD), since (DP) is a relaxation of (PD), and (OEP) is in a reduced space. The benefit of the bilevel decomposition on top of the logic-based OA, is not seen in the results presented in Table 2 due to the relatively small problem size, but becomes clear when we solve larger problems (see Examples 2-4).

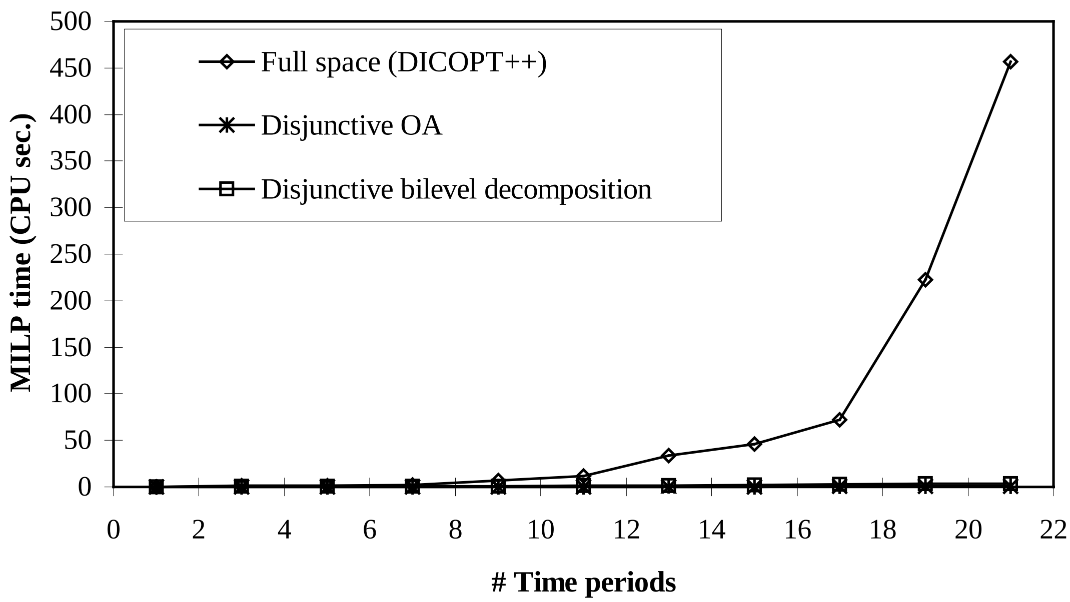

Figure 9: Comparison of MILP solution times of the proposed algorithms with DICOPT++.

Table 2: Comparison of results

Number of Time Periods	Discrete Variables	Continuous Variables	Constraints	Total solution time (CPU sec.)		
				Full Space (DICOPT++)	Disjunctive Logic-Based OA	Disjunctive Bilevel Decomposition
1	9	18	38	0.1	0.1	0.2

5	33	74	154	1.7	0.3	1.2
9	57	130	270	8.2	0.4	2.2
13	81	186	386	36.7	0.7	3.6
17	105	242	502	72.3	1.0	5.5
21	129	298	618	456.6	1.2	7.4

9.2 Example 2

Consider the superstructure (Duran and Grossmann, 1986) in Figure 10. Its formulation as problem (PD) is not given here, but is similar to the formulation of Example 1.

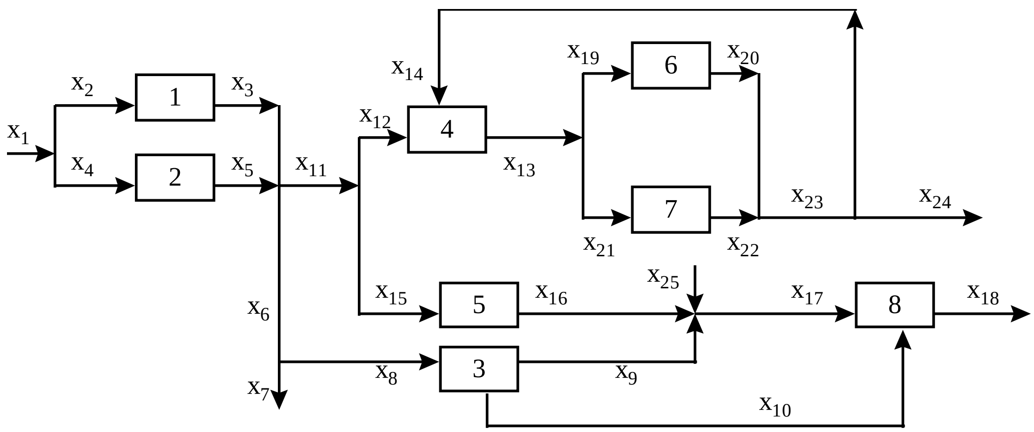

Figure 10: Eight unit superstructure**.**

We can see that the trends in the computational results obtained from example 1 hold for this larger problem. The solution times for the MILP master problems are shown in Figure 11. Note the large difference in solution times as the number of time periods increases. Table 3 also shows a comparison of the total solution time for the three methods with the corresponding problem sizes. Note the increase in the number of 0-1 variables from 24 for 1 time period to 408 for 25 time periods. For small problems (small number of time periods), the disjunctive OA algorithm seems to be most effective, while the disjunctive bilevel decomposition algorithm clearly dominates for larger problems. In particular, for the 21 period problem a reduction of nearly two orders of magnitude was achieved (156 secs. vs. 9341 secs.). Solution times are significantly reduced with the proposed algorithms, although computation time increases rapidly at larger number of periods for all three algorithms.

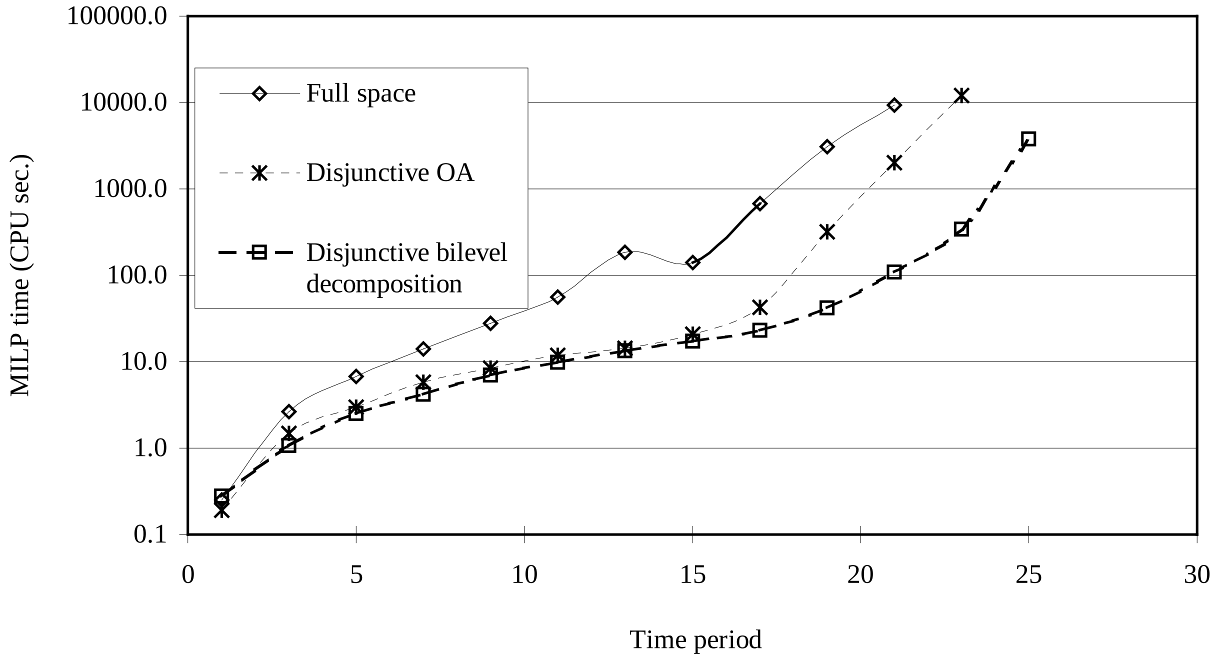

Figure 11: Comparison of algorithm performance in MILP solution time (log scale).

Table 3: Comparison of results

Number of Time Periods	Discrete Variables	Continuous Variables	Constraints	Total solution time (CPU sec.)		
				Full Space (DICOPT++)	Disjunctive Logic-Based OA	Disjunctive Bilevel Decomposition
1	24	41	142	0.5	0.5	0.8
5	88	201	614	12.3	4.4	5.4
9	152	361	1086	43.5	12.6	13.9
13	216	521	1558	222.8	20.5	30.8
17	280	681	2030	730.8	52.0	46.7
21	344	841	2502	9341.2	2047.3	156.3
23	376	921	2738	>10000	11981.8	390.8
25	408	1001	2974	>>10000	>12000	3841.7

9.3 Examples 3 and 4

Consider a multiproduct batch plant for manufacturing N products and consisting of M stages in sequence with parallel equipment in each stage.

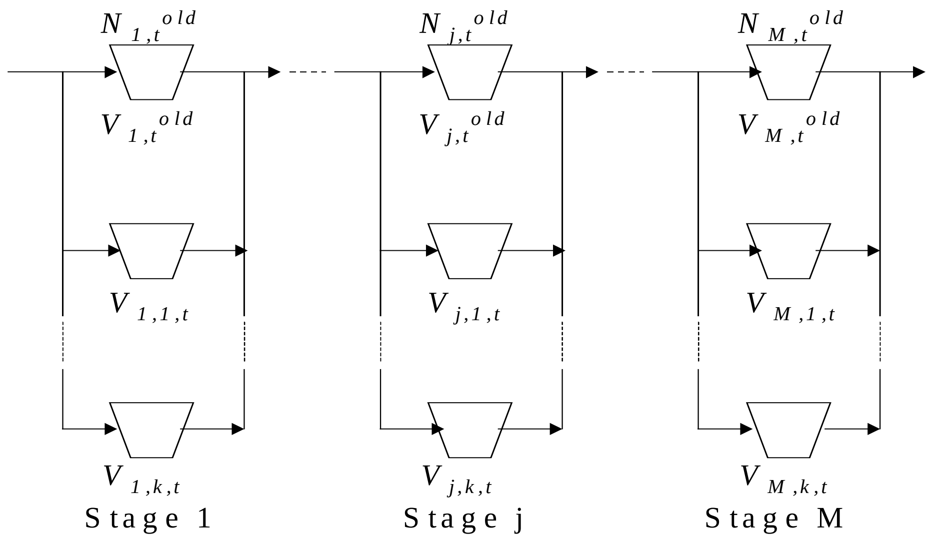

Figure 12: Superstructure for retrofit design off multiproduct batch plant (Vaselenak *et al*., 1987)

In the retrofit design of these batch plants, the problem lies in deciding the addition of new equipment to an existing plant for which new production targets and selling prices have been specified, which cannot be met by the current plant configuration (Vaselenak *et al*., 1987; Fletcher *et al*., 1991). The new units can be added to operate either in phase with an existing unit to increase its capacity (option B), or in sequence with existing units to decrease the cycle time (option C). For the multiperiod retrofit problem, new units can be added at any time period, depending on the market fluctuations that are predicted over a planning horizon. A major issue in this problem is to establish an optimal trade-off between the timing for installing new units and profit from achieving the production target by taking into account discounted costs. A revised plant configuration, operating strategy, equipment sizes and batch processing parameters for which the profit is optimal must be found in each time period.

The MINLP problem formulation of this problem for a single time period is given in Fletcher *et al*., who in contrast to Vaselenak *et al.* allowed units to operate in different modes for different products. We use the same nomenclature as Fletcher *et al*., with the difference that some of our parameters and variables are time indexed due to the multiperiod formulation. Note that expansions take place only once over the entire planning horizon, and are equivalent to the installation of a new unit. We also use here a convexified formulation of the constraints. The concave term in the objective is treated through underestimators as discussed in Vaselenak *et al*. Also, as will be shown below, the disjunctive model has a structure similar to problem (PD), except that additional disjunctions are needed to model the mode of operation in the units (option B or option C).

Nomenclature:

Parameters:

N The number of products manufactured

M The number of stages in the plant

N_j^{old} The number of existing units in stage j

$(V_j^{old})_m$ The volume of existing unit m in stage j

t_{ij} The process time of product i in stage j

H The operating time period

S_{ij} The size factor of product i in stage j

K_{jt} The annualized fixed charge of installing a new unit in stage j in period t

c_{jt} The annualized cost coefficient of installing a new unit in stage j in period t

Q_{it} The demand of product i in period t

p_{it} The expected net profit per unit of product i in period t

V_j^L The minimum volume of new units in stage j

V_j^U The maximum volume of new units in stage j

Z_j The maximum number of units which can be added to stage j

Z^U The maximum number of units which can be added to the plant

Variables:

Binary decision variables:

y_{jk} Selection of investment of unit k in stage j

w_{jkt} Operation of unit k in stage j in period t

w^B_{ijkmt} Operate new unit k in phase with existing unit m for product i in stage j in period t

w^C_{ijkt} Operate new unit k in sequence with existing units for product i in stage j in period t

z_{jkt} Expansion/installation of new unit k in stage j in period t

Continuous decision variables:

n_{it} The number of batches of product i in period t

B_{it} The batch size of product i in period t

T_{Lit} The limiting cycle time of product i in period t

V_{jkt} The volume of new unit k in stage j in period t

e_{jkt} The expansion volume of new unit k in stage j in period t

V^B_{ijkmt} The volume required in new unit k in stage j for product i to use it in phase with existing unit m in period t

V^C_{ijkt} The volume required in new unit k in stage j for product i to use it in sequence with existing units in period t

CE_{jkt} Expansion/installation cost for new unit k in stage j in period t

We define the following variables to apply the exponential transformation (Vaselenak et al., 1987): $x1_{it} = \ln n_{it}$, $x2_{it} = \ln B_{it}$, $x3_{it} = \ln T_{Lit}$. The multiperiod formulation, obtained by applying the general disjunctive model (PD), is then as follows:

i) Objective function:

$$\min\ -\sum_t \sum_i p_{it} \exp(x1_{it} + x2_{it}) + \sum_t \sum_j \sum_k CE_{tjk}$$

ii) Production targets:

$$x1_{it} + x2_{it} \le \ln Q_{it} \qquad \forall i,t$$

iii) Limiting cycle time of product i:

$$N_j^{old} + \sum_k w^C_{ijkt} \ge t_{ij} \exp(-x3_{it}) \qquad \forall i, j, t$$

iv) Yearly operating time:

$$\sum_i \exp(x1_{it} + x3_{it}) \le H_t \qquad \forall t$$

v) Bound on total number of new units:

$$\sum_j \sum_k y_{jk} \le Z^U$$

vi) Option B capacity constraints:

$$\sum_k V^B_{ijkmt} + \left(V_j^{old}\right)_m \ge S_{ij} B_{it} \qquad \forall i, j, m, t$$

vii) Distinct assignment of new units:

$$y_{jk} \ge y_{j,k+1} \qquad \forall j, k = 1 \ldots Z_j - 1$$

viii) Disjunction for every unit k added to stage j:

$$\begin{bmatrix} y_{jk} \\ V_{jkt} \le V_j^U \\ V_{jkt} = V_{jk,t-1} + e_{jkt} \\ \begin{bmatrix} w_{jkt} \\ V_{jkt} \ge V_j^L \\ \begin{bmatrix} w_{ijk1t}^B \\ V_{ijk1t}^B \le V_{jkt} \\ V_{ijk1t}^B \le V_j^U \end{bmatrix} \vee \ldots \vee \begin{bmatrix} w_{ijkmt}^B \\ V_{ijkmt}^B \le V_{jkt} \\ V_{ijkmt}^B \le V_j^U \end{bmatrix} \vee \begin{bmatrix} w_{ijkt}^C \\ V_{ijkt}^C \le V_{jkt} \\ V_{ijkt}^C \le V_j^U \\ V_{ijkt}^C \ge B_{it} S_{ij} \end{bmatrix} \forall i \\ \begin{bmatrix} z_{jkt} \\ CE_{jkt} = K_{jt} + c_{jt} e_{jkt} \\ V_j^L \le e_{jkt} \le V_j^U \end{bmatrix} \vee \begin{bmatrix} \neg z_{jkt} \\ CE_{jkt} = 0 \\ e_{jkt} = 0 \end{bmatrix} \end{bmatrix} \vee \begin{bmatrix} \neg w_{jkt} \\ V_{jkt} \ge 0 \end{bmatrix} \forall t \end{bmatrix} \vee \begin{bmatrix} \neg y_{jk} \\ V_{jkt} = 0 \end{bmatrix} \forall j,k$$

ix) Propositional logic:

$$\sum_m w_{ijkmt}^B + w_{ijkt}^C = w_{jkt} \quad \forall i, j, k, t$$

$$\sum_t z_{jkt} = y_{jk} \quad \forall j, k$$

$$y_{jk} \ge \sum_t w_{jkt} \quad \forall j, k$$

$$w_{jkt} \le y_{jk} \quad \forall j, k, t$$

$$w_{jkt} \le \sum_{\tau=1}^{t} z_{jk\tau} \quad \forall j, k, t$$

$$z_{jkt} \le w_{jkt} \quad \forall j, k, t$$

x) Variables:

$$n_{it}, B_{it}, T_{Lit}, V_{jk}, e_{jkt}, V_{jkt}, V_{ijkmt}^B, V_{ijkt}^C \ge 0 \qquad y_{jk}, w_{jkt}, w_{ijkmt}^B, w_{ijkt}^C, z_{jkt} = \{True, False\}$$

$$i = 1...N \qquad j = 1...N \qquad t = 1...T \qquad k = 1...Z_j \qquad m = 1...N_j^{old}$$

For the first time period, data for Examples 3 and 4 is the same as for Examples 1 and 2 in Fletcher *et al.* (1991) respectively, and in subsequent periods demands and costs vary. We omit this data due to the large amount of it, but interested readers can contact the authors. An example of a solution to Example 3 for four time periods is shown in Figure 13, where it can be seen that a new unit is added in stage 2 in year 1 and also in stage 2 in year 3 leading to a Net Present Value (NPV) of \$6.3 million. Furthermore, if expansions were only allowed in period 1, the

NPV would decrease by 12% to $5.6 million. This shows the value of the multiperiod formulation allowing discrete decisions for expansion and operation in each time period.

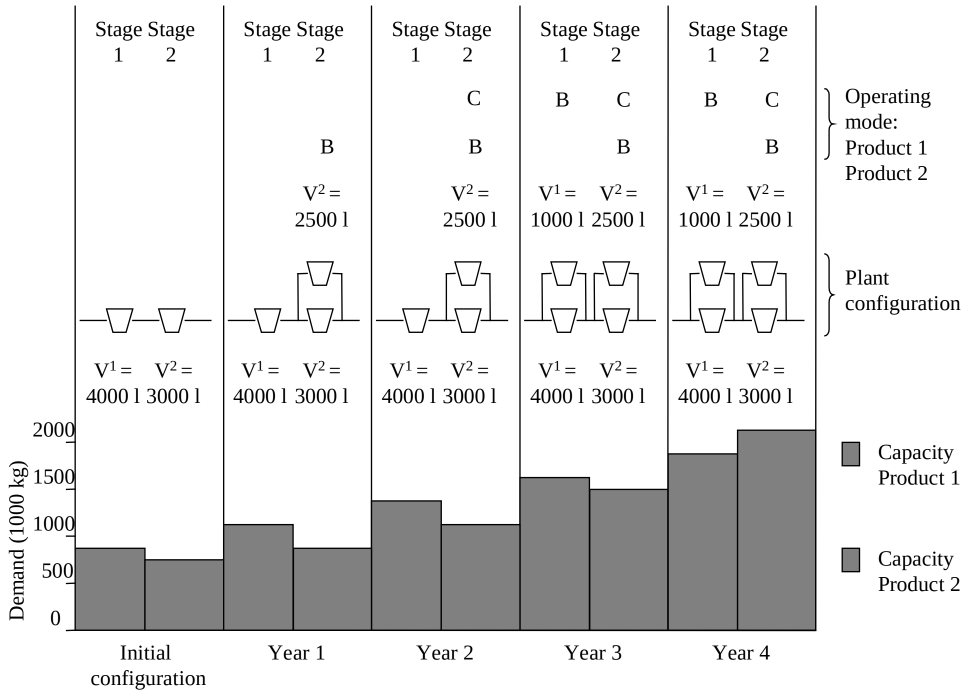

Figure 13: Solution to Example 3 for four time periods

The total solution times show the same trends as the MILP solution times, since the NLP solution times are insignificant for these examples compared to that of the MILPs. All three algorithms, i.e. the fullspace MINLP, the disjunctive OA and the disjunctive bilevel decomposition algorithm, obtain the same objective value. Once again the disjunctive bilevel decomposition algorithm dominates, with up to one or more orders of magnitude decrease in solution times compared to the full space method (see Tables 4 and 5). In Table 4 it can be seen that for the 10 period problem neither the full-space, nor the disjunctive OA, could find the optimal solution after 20 hours of computation, while the disjunctive bilevel decomposition algorithm obtained the optimal solution in less than an hour. The benefit of the disjunctive formulation is not present for the smaller problems, but is significant for larger problems as shown in Table 5.

Table 4: Comparison of results for example 3: 2 stages 2 products

Number of	Objective Profit	Discrete Variables	Continuous Variables	Constraints	MILP solution time (CPU sec.)		
					Full Space	Disjunctive	Disjunctive

Time Periods	($1000)				OA	Logic-Based OA	Bilevel Decomposition
1	1420.6	20	33	89	0.4	0.5	0.8
2	3257.7	40	65	172	2.0	2.2	1.6
4	7763.2	100	177	484	209.3	171.0	27.3
6	11911.4	180	265	722	2046.4	8372.1	81.4
8	15690.4	320	449	1250	52104.9	>50000*	718.7
10	19700.9	400	561	1560	>70000*	>70000*	3362.8

Table 5: Comparison of results for example 4: 4 stages 4 products

Number of Time Periods	Objective Profit ($1000)	Discrete Variables	Continuous Variables	Constraints	MILP solution time (CPU sec.)		
					Full Space OA	Disjunctive Logic-Based OA	Disjunctive Bilevel Decomposition
1	497.6	104	105	331	4.4	6.4	1.7
2	1041.6	312	297	936	92.4	148.2	23.7
3	1612.3	468	445	1397	>40000*	4968.4	852.6
4	2202.6	832	769	2422	>40000*	35872.1	13757.9

10. Conclusions

The algebraic method for solving multiperiod MINLPs that involve discrete decisions for topology selection, capacity expansion and operation at each time period is combinatorially explosive, as has been shown in this paper. To effectively address this problem, we have proposed a general model for design and planning of process industry networks, incorporating design, operation planning and capacity expansion in one model. Furthermore, we have proposed two algorithms - the disjunctive OA algorithm and the disjunctive bilevel decomposition algorithm - to solve this model. The proposed methods were applied to the areas of process planning and retrofit design of batch plants and show significantly reduced solution times, especially where the latter method is concerned. These algorithms specifically address the problem of the computational expense in solving the MILP step, which is often the bottleneck in the computations of multiperiod optimization problems.

* Stopped after 30 major iterations

Acknowledgment

The authors would like to acknowledge financial support from the Department of Energy under Grant number DE-FG02-85ER13396.

References

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

Brooke, A., Kendrick, D., and Meeraus, A., *GAMS: A User's Guide, Release 2.25*, The Scientific Press, South San Francisco, 1992.

Dedopoulos, I.T., and Shah, N., "Long-term Maintenance Policy Optimization in Multipurpose Process Plants," *Trans IChemE*, Vol. 74, Part A, pp. 307-320, 1996.

Duran, M.A., and Grossmann, I.E., "An Outer Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs," *Math. Program.*, Vol. 36, pp. 307-339, 1986.

Fletcher, R., Hall, J.A.J., and Johns, W.R., "Flexible Retrofit Design of Multiproduct Batch Plants," *Computers chem. Engng*, Vol. 15, No. 12, pp. 843-852, 1991.

Garey, M.R., and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, New York, 1978.

Geoffrion, A.M., "Generalized Benders Decomposition," *J. Optim. Theory Appl.*, Vol. 10, p. 237, 1972.

Iyer, R.R., and Grossmann, I.E., "Optimal Multiperiod Operational Planning for Utility Systems," *Computers chem. Engng*, Vol. 21, No. 8, pp. 787-800, 1997.

Iyer, R.R., and Grossmann, I.E., "A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks," *Ind. Eng. Chem. Res.*, Vol. 37, No. 2, pp. 474-481, 1998.

Kocis, G.R. and Grossmann, I.E., "A modeling and decomposition strategy for the MINLP optimization of process flowsheets," *Computers chem. Engng*, Vol. 13, pp. 797-819, 1989.

Papalexandri, K.P., and Pistikopoulos, E.N., "A Multiperiod MINLP Model for the Synthesis of Flexible Heat and Mass Exchange Networks," *Computers chem. Engng*, Vol. 18, No. 11/12, pp. 1125-1139, 1994.

Paules, G.E. IV, and Floudas, C.A., "Stochastic Programming in Process Synthesis: A Two-Stage Model with MINLP Recourse for Multiperiod Heat-Integrated Distillation Sequences," *Computers chem. Engng*, Vol. 16, No. 3, pp. 189-210, 1992.

Quesada, I., and Grossmann, I.E., "An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems," *Computers chem. Engng*, Vol. 16, pp. 937-947, 1992.

Raman, R., and Grossmann, I.E., "Modeling and Computational Techniques for Logic Based integer Programming," *Computers chem. Engng*, Vol. 18, No. 7, pp. 563-578, 1994.

Sahinidis, N.V., Grossmann, I.E., Fornari, R.E., and Chathrathi, M., "Optimization Model for Long Range Planning in the Chemical Industry," *Computers chem. Engng*, Vol. 13, No. 9, pp. 1049-1063, 1989.

Sahinidis, N.V., and Liu, M., "Long Range Planning in the Process Industries: A Projection Approach," *Comput. Oper. Res.*,Vol. 23, No. 3, pp. 237-253, 1996.

Turkay, M., and Grossmann, I.E., "Logic-Based MINLP Algorithms for the Optimal Synthesis of Process Networks," *Computers chem. Engng*, Vol. 20, No. 8, pp. 959-978, 1996a.

Turkay, M., and Grossmann, I.E., "Disjunctive Programming Techniques for the Optimization of Process Systems with Discontinuous Investment Costs – Multiple Size Regions," *Ind. Ebg. Chem. Res.*, Vol. 35, pp. 2611-2623, 1996b.

Varvarezos, D.K., Grossmann, I.E., and Biegler, L.T., "An Outer-Approximation Method for Multiperiod design Optimization," *Ind. Eng. Chem. Res.*, Vol. 31, No. 6, pp. 1466-1477, 1992.

Vaselenak, J.A., Grossmann, I.E., and Westerberg, A.W., "Optimal Retrofit Design in Multiproduct Batch Plants," *Ind. Engng Chem. Res.* Vol. 26, pp. 718-726, 1987.

Appendix

Appendix A: Derivation of the convex hull of problem (DMILP)

Raman and Grossmann (1994) showed how to convert linear disjunctive programs to mixed integer form through the convex hull formulation for each disjunction, based on previous work by Balas (1985). We use the same ideas to convert problem (DLP) to problem (DMILP) through the convex hull formulation for each disjunction. The basic idea is to replace the Boolean variables by corresponding binary variables and to disaggregate the continuous variables to have a variable for each disjunction. The convex hull gives a tighter formulation than the "big-M" formulation (Turkay and Grossmann, 1996b). As will be seen in the following derivation, a large number of the disaggregated variables become redundant for this model, and can be removed to simplify the model. Consider problem (DLP):

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

$$g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \le 0 \quad \forall t$$
$$Dx_t \le 0 \qquad \forall t$$

$$\left[\begin{array}{c} y_j \\ \left[\begin{array}{c} w_{jt} \\ h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T (x_{t,nz} - x_{t,nz}^k) + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T (x_{t,z} - x_{t,z}^k) \le a \\ Ax_{t,nz} + Bx_{t,z} + CQ_{jt} \le b \\ k \in K^j \\ CO_{jt} = \gamma_{jt} \\ \left[\begin{array}{c} z_{jt} \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array}\right] \vee \left[\begin{array}{c} \neg z_{jt} \\ Q_{jt} = Q_{j,t-1} \\ CE_{jt} = 0 \end{array}\right] \end{array}\right] \vee \left[\begin{array}{c} \neg w_{jt} \\ B^{jt} x_t = 0 \\ CO_{jt} = 0 \end{array}\right] \forall t \end{array}\right] \vee \left[\begin{array}{c} \neg y_j \\ B^{jt} x_t = 0 \\ \forall t \end{array}\right] \quad \forall j$$

$$y_j \Rightarrow \bigvee_{t=1}^{T} w_{jt} \quad \forall j \qquad\qquad w_{jt} \Rightarrow y_j \quad \forall j,t$$
$$w_{jt} \Rightarrow \bigvee_{\tau=1}^{t} z_{jt} \quad \forall j,t \qquad\qquad z_{jt} \Rightarrow w_{jt} \quad \forall j,t$$
$$\Omega(y) = True$$
$$CE, CO, Q, QE, x \ge 0, \qquad y, w, z \in \{True, False\}$$

To convert each disjunction to mixed-integer form through the convex hull formulation, we first convert the inner disjunction and work outwards until the whole problem is transformed into an MILP problem. For simplicity, we will ignore most of the sub- and superscripts for the rest of the derivation, except where they are relevant. First, consider the innermost disjunction:

$$\left[\begin{array}{c} z_{jt} \\ CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} \end{array}\right] \vee \left[\begin{array}{c} \neg z_{jt} \\ CE_{jt} = 0 \\ QE_{jt} = 0 \end{array}\right] \tag{A1}$$

The expansion equations can in this case be moved to the outer disjunction to simplify derivation of the convex hull, seeing that they always apply if the process is chosen. The first step is to disaggregate the variables. Superscript 1 refers to the left disjunction, while superscript 2 refers to the right disjunction. A binary variable is assigned to every disjunction. The disaggregation is then as follows:

$$\begin{aligned} & z^1 + z^2 = 1 \\ & QE = QE^1 + QE^2 \\ & CE = CE^1 + CE^2 \end{aligned} \tag{A2}$$

Substitute disaggregated variables into disjunction 1:

$$CE^1 = \alpha QE^1 + \beta z^1 \qquad (A3)$$

$$0z^1 \le QE^1 \le Uz^1 \qquad (A4)$$

Substitute disaggregated variables into disjunction 2:

$$CE^2 = 0z^2 = 0 \qquad (A5)$$

$$QE^2 = 0z^2 = 0 \qquad (A6)$$

Simplify:

$$\begin{aligned} &CE = \alpha QE + \beta z \\ &QE \le Uz \\ &QE, CE \ge 0 \\ &z \in \{0,1\} \end{aligned} \qquad (A7)$$

Next consider the middle disjunction in (DLP) after replacing the inner disjunction with (A7):

$$\begin{bmatrix} w_{jt} \\ h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T (x_{t,nz} - x_{t,nz}^k) + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T (x_{t,z} - x_{t,z}^k) \le a \\ Ax_{t,nz} + Bx_{t,z} + CQ_{jt} \le b \\ k \in K^j \\ CO_{jt} = \gamma_{jt} \\ (A7) \end{bmatrix} \vee \begin{bmatrix} \neg w_{jt} \\ B^{jt} x_{t,z} = 0 \\ CO_{jt} = 0 \end{bmatrix} \qquad (A8)$$

The same costs and continuous variables also go to zero in the right hand side disjunction as the ones in (A7), but they are left out for simplicity. After a similar procedure is applied as in the derivation of (A7), namely disaggregation of the continuous variables, substitution and simplification, the following mixed integer convex hull formulation is obtained:

$$x_{t,nz} = x^1_{t,nz} + x^2_{t,nz}$$

$$Q_{jt} = Q^1_{jt} + Q^2_{jt}$$

$$\nabla_{x_{t,nz}} h^{nl}_{jt}(x^k_t)^T x^1_{t,nz} + \nabla_{x_{t,z}} h^{nl}_{jt}(x^k_t)^T x_{t,z} \le \left[-h^{nl}_{jt}(x^k_t) + \nabla_{x_{t,nz}} h^{nl}_{jt}(x^k_t)^T x^k_{t,nz} + \nabla_{x_{t,z}} h^{nl}_{jt}(x^k_t)^T x^k_{t,z} + a \right] w_{jt}$$

$$Ax^1_{t,nz} + Bx_{t,z} + CQ^1_{jt} \le bw_{jt}$$

$$CO_{jt} = \gamma_{jt} w_{jt}$$

$$CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt}$$

$$QE_{jt} \le Uz_{jt}$$

$$x_{t,z} \le Uw_{jt}$$

$$x^1_{t,nz} \le Uw_{jt}$$

$$Q^1_{jt} \le Uw_{jt}$$

$$x^2_{t,nz} \le U(1 - w_{jt})$$

$$Q^2_{jt} \le U(1 - w_{jt})$$

$$z_{jt} \le w_{jt}$$

$$x, Q, QE, CE, CO \ge 0, \qquad z, w \in \{0,1\} \qquad \text{(A9)}$$

Finally consider the outer disjunction in (DLP) with (A9):

$$\begin{bmatrix} y_j \\ Q_{jt} = Q_{j,t-1} + QE_{jt} \\ (A9) \end{bmatrix} \vee \begin{bmatrix} \neg y_j \\ B^{jt} x_t = 0 \\ Q_{jt} = 0 \end{bmatrix}$$

Although it is not shown above, the same variables are set to zero on the right hand side disjunction as in the formulation of (A9), but they are set to zero for all time periods. After disaggregation of variables, substitution and simplification, the constraints outside the disjunctions are added and logical propositions are expressed in algebraic form to give the final (MILP) formulation, model (DMILP):

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \le 0 \quad \forall t$$

$$Dx_t \le d \quad \forall t$$

$$\nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^1 + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z} \le -\left[h_{jt}^{nl}(x_t^k) - \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^k + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z}^k - a\right] w_{jt}$$

$$\forall j, t, k \in K^j$$

$$Ax_{t,nz}^1 + Bx_{t,z} + CQ_{jt}^1 \le bw_{jt} \quad \forall j, t$$

$$x_{t,nz} = x_{t,nz}^1 + x_{t,nz}^2 \quad \forall t$$

$$x_{t,nz}^1 \le Uw_{jt} \quad \forall j, t$$

$$x_{t,nz}^2 \le U(1 - w_{jt}) \quad \forall j, t$$

$$x_{t,z} \le Uw_{jt} \quad \forall j, t$$

$$Q_{jt} = Q_{jt}^1 + Q_{jt}^2 \quad \forall j, t$$

$$Q_{jt}^1 \le Uw_{jt} \quad \forall j, t$$

$$Q_{jt}^2 \le U(1 - w_{jt}) \quad \forall j, t$$

$$CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} \quad \forall j, t$$

$$CO_{jt} = \gamma_{jt} w_{jt} \quad \forall j, t$$

$$Q_{jt} = Q_{j,t-1} + QE_{jt} \quad \forall j, t$$

$$QE_{jt} - Uz_{jt} \le 0 \quad \forall j, t$$

$$y_j \le \sum_t w_{jt} \quad \forall j$$

$$w_{jt} \le y_j \quad \forall j, t$$

$$w_{jt} \le \sum_{\tau=1}^{t} z_{j\tau} \quad \forall j, t$$

$$z_{jt} \le w_{jt} \quad \forall j, t$$

$$Ey \le e$$

$$CE, CO, Q, QE, x \ge 0, \qquad y, w, z \in \{0,1\}$$

Appendix B: Derivation of the convex hull for the MILP master problems in the bilevel decomposition

Upper level design problem:

The convex hull for this problem is the same as (DMILP), except that the design variable, *y*, is used to drive variables to zero, and the binary variables for operation and expansion planning are relaxed using either integer relaxation or the following relaxations:

$$w_{jt} \geq \frac{x_t}{x_{t,upper}} \qquad (B1)$$

$$w_{jt} \geq \frac{x_t}{x_t^k} \qquad k \in K \qquad (B2)$$

$$z_{jt} \geq \frac{QE_{jt}}{QE_{jt,upper}} \qquad (B3)$$

$$z_{jt} \geq \frac{QE_{jt}}{QE_{jt}^k} \qquad k \in K \qquad (B4)$$

where K is the number of major iterations between (DP) and (OEP). For (B1) and (B3), upper bounds are used, while the values obtained in the previous (OEP) problem are used in (B2) and (B4). The values on the right hand sides of the expressions above will always be less or equal than 1, and this is therefore a relaxation of the original problem.

After substituting these relaxations, the MILP master problem for (DP) is as follows (MIPDP):

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \le 0 \quad \forall t$$

$$Dx_t \le d \quad \forall t$$

$$\nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^1 + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z} \le -\left[h_{jt}^{nl}(x_t^k) - \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^k + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z}^k - a\right] y_j$$

$$\forall j,t \quad k \in K^j$$

$$Ax_{t,nz}^1 + Bx_{t,z} + CQ_{jt} \le by_j \quad \forall j,t$$

$$x_{t,nz} = x_{t,nz}^1 + x_{t,nz}^2 \quad \forall t$$

$$x_{t,nz}^1 \le Uy_j \quad \forall j,t$$

$$x_{t,nz}^2 \le U(1 - y_j) \quad \forall j,t$$

$$x_{t,z} \le Uy_j \quad \forall j,t$$

$$CE_{jt} \ge \alpha_{jt} QE_{jt} + \beta_{jt} \frac{QE_{jt}}{QE_{jt,upper}} \quad \forall j,t$$

$$CE_{jt} \ge \alpha_{jt} QE_{jt} + \beta_{jt} \frac{QE_{jt}}{QE_{jt}^k} \quad \forall j,t$$

$$CO_{jt} = \gamma_{jt} \frac{x_t}{x_{t,upper}} \quad \forall j,t$$

$$CO_{jt} = \gamma_{jt} \frac{x_t}{x_t^k} \quad \forall j,t$$

$$Q_{jt} = Q_{j,t-1} + QE_{jt} \quad \forall j,t$$

$$QE_{jt} - Uy_j \le 0 \quad \forall j,t$$

$$Ey \le e$$

$$CE, CO, Q, QE, x \ge 0, \quad y \in \{0,1\}$$

For the lower level planning problem, the formulation is the same as for (A9), except that the y values are fixed and with the addition of the expansion equations. The MILP master formulation is as follows (MIPOEP):

$$\min \quad \sum_t \sum_j CO_{jt} + \sum_t \sum_j CE_{jt} + \sum_t \sum_i c_{it} x_{it}$$

subject to

$$g_t^{nl}(x_t^k) + \nabla_{x_t} g_t^{nl}(x_t^k)^T (x_t - x_t^k) \le 0 \quad \forall t$$

$$Dx_t \le d \quad \forall t$$

$$x_{t,nz} = x_{t,nz}^1 + x_{t,nz}^2 \quad \forall t$$

$$Q_{jt} = Q_{jt}^1 + Q_{jt}^2 \quad \forall j,t$$

$$\nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^1 + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z} \le \left[-h_{jt}^{nl}(x_t^k) + \nabla_{x_{t,nz}} h_{jt}^{nl}(x_t^k)^T x_{t,nz}^k + \nabla_{x_{t,z}} h_{jt}^{nl}(x_t^k)^T x_{t,z}^k + a \right] w_{jt}$$

$$\forall j,t,k \in K^j$$

$$Ax_{t,nz}^1 + Bx_{t,z} + CQ_{jt}^1 \le bw_{jt} \quad \forall j,t$$

$$CO_{jt} = \gamma_{jt} w_{jt} \quad \forall j,t$$

$$CE_{jt} = \alpha_{jt} QE_{jt} + \beta_{jt} z_{jt} \quad \forall j,t$$

$$Q_{jt} = Q_{j,t-1} + QE_{jt} \quad \forall j,t$$

$$QE_{jt} \le Uz_{jt} \quad \forall j,t$$

$$x_{t,z} \le Uw_{jt} \quad \forall j,t$$

$$x_{t,nz}^1 \le Uw_{jt} \quad \forall j,t$$

$$Q_{jt}^1 \le Uw_{jt} \quad \forall j,t$$

$$x_{t,nz}^2 \le U(1 - w_{jt}) \quad \forall j,t$$

$$Q_{jt}^2 \le U(1 - w_{jt}) \quad \forall j,t$$

$$y_{j,fixed} \le \sum_t w_{jt} \quad \forall j$$

$$w_{jt} \le y_{j,fixed} \quad \forall j,t$$

$$w_{jt} \le \sum_{\tau=1}^{t} z_{j\tau} \quad \forall j,t$$

$$z_{jt} \le w_{jt} \quad \forall j,t$$

$$Fw \le f$$

$$x, Q, QE, CE, CO \ge 0, \qquad z, w \in \{0,1\}$$