# An Iterative Aggregation/Disaggregation Approach for the Solution of a Mixed Integer Nonlinear Oilfield Infrastructure Planning Model

Susara A. van den Heever* and Ignacio E. Grossmann**
Department of Chemical Engineering, Carnegie Mellon University
Pittsburgh, PA, 15213
August 1999

## Keywords
Oilfield planning, MINLP, aggregation, decomposition

## Abstract

A multiperiod MINLP model for offshore oilfield infrastructure planning is presented where nonlinear reservoir behavior is incorporated directly into the formulation. Discrete decisions include the selection of production platforms, well platforms and wells to be installed/drilled, as well as the drilling schedule for the wells over the planning horizon. Continuous decisions include the capacities of the platforms, as well as the production profile for each well in each time period.

For the solution of this model, an iterative aggregation/disaggregation algorithm is proposed in which logic-based methods, a bilevel decomposition technique, the use of convex envelopes and aggregation of time periods are integrated. Furthermore, a novel dynamic programming sub-problem is proposed to improve the aggregation scheme at each iteration in order to obtain an aggregate problem that resembles the disaggregate problem more closely. A number of examples are presented to illustrate the performance of the proposed method.

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

## Introduction

Offshore oilfield infrastructure planning is a challenging problem encompassing both complex physical constraints and intricate economical specifications. An offshore oilfield infrastructure consists of Production Platforms (PP), Well Platforms (WP), wells and connecting pipelines (see Figure 1), and is constructed for the purpose of producing oil and/or gas from one or more oilfields. 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.

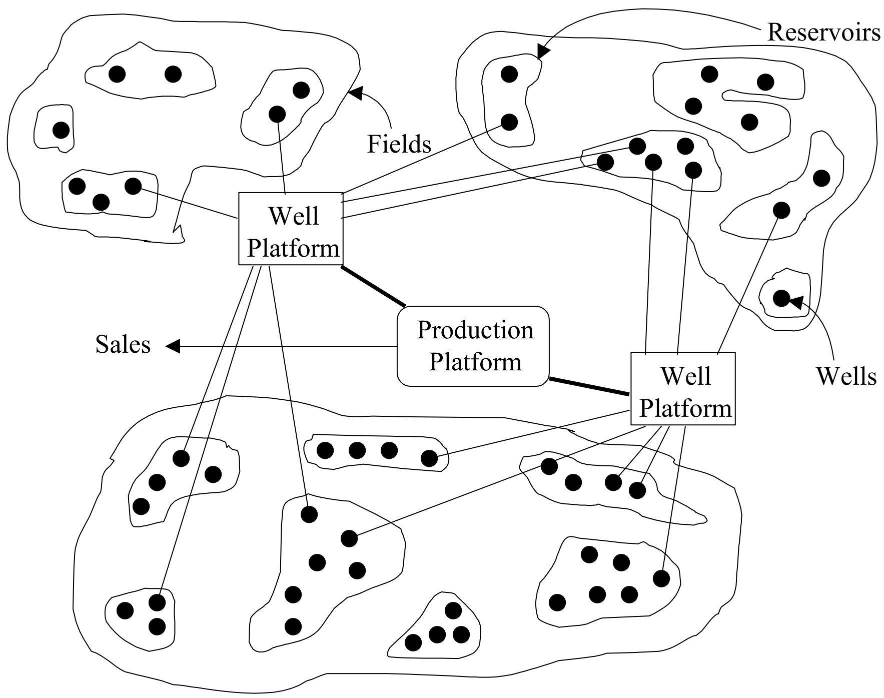


**Figure 1:** Configuration of fields, well platforms and production platforms (Iyer *et al.*, 1998)

Offshore oilfield facilities are often in operation over several decades and it is therefore important to take future conditions into consideration when designing an initial infrastructure. This can be incorporated by dividing the operating horizon, for example 30 years, into a number of time periods, for example 30 periods with a length of 1 year each, and allowing planning decisions in each period, while design decisions are made for the horizon as a whole.

Design decisions involve the capacities of the PPs and WPs, as well as decisions regarding *which* PPs, WPs and wells to install over the whole operating horizon. Planning decisions involve the production profiles in each period, as well as decisions regarding *when* to install PPs, WPs, and wells included in the design. Decision variables can also be grouped into discrete variables, for example those representing the installation of PPs, WPs and wells in each period, and continuous variables, for example those representing the production profiles and pressures in each period. This discrete/continuous mixture, together with the nonlinear reservoir behavior, requires a Mixed Integer Nonlinear Programming (MINLP) model, where the objective is to maximize or minimize a specific value function. MINLPs are known to be NP-complete (Garey and Johnson, 1978), and when the models are multiperiod, their solution easily becomes intractable. Specialized techniques are therefore often needed to solve these models. In the case of oilfield infrastructure planning, MINLP models have traditionally been avoided in favor of Mixed Integer Linear Programs (MILPs) or Linear Programs (LPs), due to the difficulties associated with dealing directly with nonlinearities and, in the latter case, discrete decisions. It is important to note that specialized algorithms may be necessary even when nonlinearities are avoided, due to the large magnitude of the multiperiod optimization problem.

In the past, decisions regarding capacities of platforms, drilling schedules and production profiles have often been made separately with certain assumptions to ease the computational burden. One such approach is to assume a fixed drilling schedule and then to determine the production profile by solving an LP (e.g. Lee and Aranofsky, 1958). Another is to fix the production profile in order to determine a well drilling schedule by solving an LP and subsequently rounding non-integer solutions to reasonable integer values (Aranofsky and Williams, 1961; Attra *et al*., 1960), or by solving an MILP. Frair (1973) proposed independent models for determining the number of PPs, capacities of platforms, and the well drilling schedule. The unfortunate consequence of not including all the design and planning decisions in one model, is that important interactions between these variables are not taken into account, leading to infeasible or suboptimal solutions.

Simultaneous models have only emerged recently. Sullivan (1982) developed a simultaneous MILP model for gasfield design and production planning, while Bohannon (1970) and Haugland *et al.* (1988) proposed simultaneous MILP models for oilfield design and production planning. All three authors use commercial MILP solvers to obtain solutions. Haugland *et al.* demonstrate that their model can only be solved for very small instances due to the computational burden, thus reiterating the need for specialized algorithms. Nygreen *et al.* (1998) present an updated version of the model proposed by Haugland *et al.* and report on its practical use over the past 15 years. Eeg and Herring (1997) combined the approach of assuming a production profile to obtain a drilling schedule and the approach of fixing the drilling schedule to obtain a production profile into an iterative scheme. In this scheme, production profiles are updated after each iteration, and convergence is achieved when the drilling schedules from the two subproblems match each other. Iyer *et al.* (1998) proposed a multiperiod MILP model for the planning and scheduling of investment and operation of offshore facilities. To be able to solve realistic problem sizes, the authors propose a sequential decomposition strategy using aggregation of time periods and wells, followed by successive disaggregation.

In all the models mentioned in the previous paragraph, the nonlinear reservoir behavior was approximated by one or more linear constraints. Eeg and Herring perform reservoir simulations to obtain production profiles for each well, and these production profiles are then linearized into discrete differences and used as input into the MILP model. Haugland *et al.* and Lee and Aranofsky assume ideal liquid reservoir behavior and derive linear constraints for this case. However, this assumption is not always realistic, since reservoirs often contain a substantial volume of gas in which case a single linear constraint will be too inaccurate if pressures vary over a large interval. Sullivan, as well as Iyer *et al.*, approximate nonlinear reservoir performance by using piecewise linear interpolation using integer variables. The benefit of this approach is improved accuracy, but this might be offset by the computational effort resulting from the large increase in integer variables. It is therefore important to deal with nonlinearities directly, not only to avoid the large number of integer variables associated with piecewise linear interpolation, but also to reduce the loss of accuracy and improve the quality of the solution.

Dawson and Fuller (1996) formulated an MINLP model assuming that the PPs, WPs and well configuration are fixed, and only considering the periodical replacement of pumps at well sites as discrete decisions. The authors include nonlinear reservoir behavior as a log-linear relationship between oil fraction and cumulative oil produced, but they do not include nonlinearities that arise from surface pressure constraints. To solve their model, they proposed a heuristic method based on Generalized Benders Decomposition (GBD, Geoffrion, 1972), coded in the GAMS modeling language (Brooke *et al*., 1992). Only limited sized problems could be solved (up to 3 wells) due to the computational burden.

In this paper we address the formulation and solution of a discrete nonlinear optimization model for oilfield planning. Our goal is to develop an algorithm capable of (1) dealing with nonlinearities directly, and (2) solving realistic instances of nonlinear offshore oilfield infrastructure design and planning models in reasonable time. The model we consider to this end, is based on the one proposed by Iyer *et al*., and includes decisions regarding investment of PPs, WP, and wells, as well as the well drilling schedule and production profile in each time period. While Iyer *et al*. (1998) used linear interpolation to approximate nonlinearities, we include these nonlinearities directly into the model. Specifically, these are the reservoir pressures, gas to oil ratio, and cumulative gas produced expressed as nonlinear functions of the cumulative oil produced.

Two techniques that play a significant role in dealing with the large size of multiperiod problems, are decomposition and aggregation. A number of papers proposing decomposition algorithms for multiperiod MINLPs in chemical engineering applications have been published (Varvarezos *et al*., 1992; Papalexandri and Pistikopoulos, 1994; Paules and Floudas, 1992; Van den Heever and Grossmann, 1999). Our approach is based on the disjunctive bilevel decomposition algorithm proposed by Van den Heever and Grossmann (1999). This algorithm involves decomposition of the design and planning model into an upper level Design Problem (DP), in which the planning decisions are relaxed, and a lower level Operation and Expansion Planning problem (OEP), which is solved for the fixed configuration obtained in (DP).

We propose and iterative aggregation/disaggregation algorithm, where the time periods are aggregated in the design problem, and subsequently disaggregated when the planning problems is solved for the fixed infrastructure obtained from the design problem. The solution from the planning problem is used to update the aggregation scheme after each iteration. This is done through a dynamic programming subproblem which determines how time periods should be aggregated to yield an aggregate problem that resembles the disaggregate problem as close as possible. Convex envelopes are used to deal with non-convexities arising from non-linearities. A number of examples are presented to show the effectiveness of the proposed algorithm.

## Problem Statement

In this paper, we consider the design and planning of an offshore oilfield infrastructure (refer to Figure 1) over a planning horizon of Y years, divided into T time periods (e.g. quarterly time periods) as discussed by Iyer *et al*. An oilfield layout consists of a number of fields, each containing one or more reservoirs and each reservoir contains one or more wellsites. After the decision has been made to produce oil from a given wellsite, it is drilled from a WP using drilling rigs. A network of pipelines connects the wells to the WPs and the WPs to the PPs. For our purposes, we assume that the location/allocation problem has been solved, i.e. the possible locations of the PPs and WPs, as well as the assignment of wells to WPs and WPs to PPs, are fixed. The design decisions we consider are valid for the whole planning horizon and are:

a) whether or not to include each PP, WP, and well in the infrastructure over the planning horizon
b) the capacities of the PPs and WPs

The planning decisions are made in each time period and are:

a) whether or not to install each PP and WP, and whether or not to drill each well
b) the production profile

These decisions are made under the assumption that operating conditions are constant during a given time period.

## Model

### Background

Iyer *et al.* stated that the primary issues involved in the problem are the reservoir behavior as a function of time and the surface pressure constraints. These authors give an elaborate description of the reservoir behavior, and we will therefore not go into too much detail here. Instead, we will concentrate on the nonlinear behavior. The productivity index and the reservoir pressure determine the oil production rate from a well in a given time period, and the latter is nonlinearly related to the cumulative oil produced. The well is usually capped when the GOR (gas to oil ratio) exceeds a certain threshold limit or when the pressure of the reservoir is lower than a minimum pressure. Iyer *et al*. modeled these nonlinearities by using linear interpolation between a set of data points obtained from reservoir simulations. This led to a large increase in the number of binary variables in the overall optimization model (see Table 1), thereby increasing the computational effort required for the branch and bound search.

**Table 1**. Growth in problem size with linear interpolation

| | **Discrete variables** | **Continuous variables** | **Constraints** |
|---|---|---|---|
| **Without linear interpolation** | 1500 | 5500 | 10000 |
| **With linear interpolation** | 5000 | 10000 | 18000 |

We propose the disjunctive model (P) in the next section, which is based on the model proposed by Iyer et al. A major difference in our model is that we use nonlinear constraints directly, thereby eliminating the need for linear interpolation. These constraints are obtained by fitting a nonlinear curve to the data points obtained from a reservoir simulation. The other major differences are that we do not include the scheduling of drilling rigs in our model, and use disjunctive programming to formulate the model.

Raman and Grossmann (1994) developed a logic-based formulation for mixed-integer programming denoted as generalized disjunctive programming. As an illustrative example in the context of oilfield planning, 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 (¬):

$$\begin{bmatrix} \bigvee_{\theta=1}^{t} z_\theta^w \\ h_t^w(x) \le 0 \\ c_t^w = \alpha_t^w \end{bmatrix} \vee \begin{bmatrix} \neg \bigvee_{\theta=1}^{t} z_\theta^w \\ B_t^w x = 0 \\ c_t^w = 0 \end{bmatrix} \qquad \forall w,t$$

In the above representation, the variable $z_t^w$ is a Boolean variable, which is true if well *w* is drilled in period *t*. Thus, the disjunction can be interpreted as follows: If well *w* has been drilled in or before time period *t* (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^w$ = true), i.e. if oil is produced from well *w* in period *t*, then enforce constraint $h_t^w(t)$ describing that well characteristics and apply a fixed cost $\alpha_t^w$. If well *w* has not been drilled in period *t* or in any period before period *t* (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^w$ = false), i.e. oil is not produced from well *w* in period *t*, the fixed cost and a subset of continuous variables are set to zero through the matrix $B_t^w$. The logical relationships among discrete variables describing connections and interactions between wells and platforms are given through logic propositions consisting of the variables and the logic operators (such as OR (∨), AND (∧), NOT (¬) and IMPLY (⇒)). For example,

$$\underline{\bigvee}_{t=1}^{T} z_t^w$$

means that each well can be drilled only once. What follows is a general formulation of the disjunctive model. The detailed model is given in Appendix A.

**Disjunctive Model (P)**

Sets and Indices:

| | |
|---|---|
| $PP$ | set of production platforms |
| $p$ | production platform $p \in PP$ |
| $WP(p)$ | set of well platforms associated with platform $p$ |
| $\pi$ | well platform $\pi \in WP(p)$ |
| $W_{WP}(\pi)$ | set of wells associated with well platform $\pi$ |
| $w$ | well $w \in W_{(.)}(.)$ |
| $T$ | set of time periods |
| $t$ | time period $t \in T$ |

Variables:

| | |
|---|---|
| $x$ | continuous variables |
| $z_t$ | discrete variables for installation/drilling of well, WP or PP in time period $t$ |

$$\max \Psi(x, z) \qquad (1)$$

subject to

$$Ax_t \leq b \qquad \forall t = 1..T \qquad (2)$$

$$f_t(x_t) \leq 0 \qquad \forall t = 1..T \qquad (3)$$

$$g_t(x_1, x_2..x_t) \leq 0 \qquad \forall t = 1..T \qquad (4)$$

$$
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_{\theta}^{p} \\
A_p x_t^p \le b_p \;(5) \\
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p} \\
A_\pi x_t^{\pi,p} \le b_\pi \;(7) \\
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_{\theta}^{w,\pi,p} \\
A_w x_t^{w,\pi,p} \le b_w \;(9) \\
h_{t,w}(x_t^{w,\pi,p}) \le 0 \;(10) \\
g_{t,w}(x_1^{w,\pi,p}, x_2^{w,\pi,p} .. x_t^{w,\pi,p}) \le 0 \;(11)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_{\theta}^{w,\pi,p} \\
B^{w,\pi,p,t} x_t^{w,\pi,p} = 0 \;(12)
\end{array}\right] \\
\forall w \in W_{WP}(\pi)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p} \\
B^{\pi,p,t} x_t^{\pi,p} = 0 \;(8)
\end{array}\right] \\
\forall \pi \in WP(p)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_{\theta}^{p} \\
B^{p,t} x_t^{p} = 0 \;(6)
\end{array}\right]
$$

$$\forall p \in PP, t \in T$$

(P)

$$\Omega(z) = True \qquad (13)$$
$$x \ge 0, \quad z \in \{True, False\}$$

The objective, represented by (1), is to maximize the Net Present Value (NPV), which includes the revenues from sale of oil, investment cost and depreciation of facilities. (2) represents linear constraints that are valid for the complete infrastructure and are not associated with any specific well or platform, such as total production balances. (3) represents nonlinear constraints that are not associated with any specific well, WP or PP, namely reservoir pressure as a function of the cumulative oil produced from that reservoir (see Figure 2(a)). Linear linking constraints that are valid for the whole infrastructure and that link the time periods together preventing a solution procedure where each time period is solved separately are represented by (4).

The outer disjunction is valid for each PP in each time period and can be interpreted as follows: If production platform $p$ has been installed during or before period $t$ (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{p}$ = true), then all constraints in the largest bracket are applied. Otherwise, if the production platform has not been installed yet (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{p}$ = false), a subset of variables associated with that PP are set to zero in (6). The matrix $B$ denotes which variables to set to zero, for example oil flowrates. The middle disjunction is valid for all well platforms associated with production platform $p$ and is only applied if the discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{p}$ is true. This disjunction states that if well platform π has been installed before or during period $t$ (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p}$ = true), then (7) is applied. Otherwise, if the well platform has not been installed (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p}$ = false), a subset of variables associated with that WP are set to zero in (8). (7) represents linear constraints associated with a specific WP, for example mass balances and pressure balances.

The innermost disjunction is valid for each well $w$ associated with well platform $\pi$, and is only included if $\pi$ has already been installed (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p}$ = true). If well $w$ has been drilled during or before period t (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{w,\pi,p}$ = true), then (9),(10) and (11) are applied. Otherwise, if well $w$ has not been drilled yet (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{w,\pi,p}$ = false), a subset of variables associated with that well are set to zero in (12). (9) denotes linear constraints associated with each well, for example mass balances and pressure balances, while (10) represents nonlinear constraints associated with each well, namely the nonlinear relationships of gas-to-oil ratio (GOR) and cumulative gas produced versus cumulative oil produced (see Figures 2(b) and 2(c)). Linear linking constraints that are valid for each well and that link the time periods together preventing a solution procedure where each time period is solved separately, are represented by (11). Finally, constraint (13) represents logical relationships between the discrete decisions, for example, only three wells can be drilled per time period and each well, WP or PP can only be drilled/installed once.

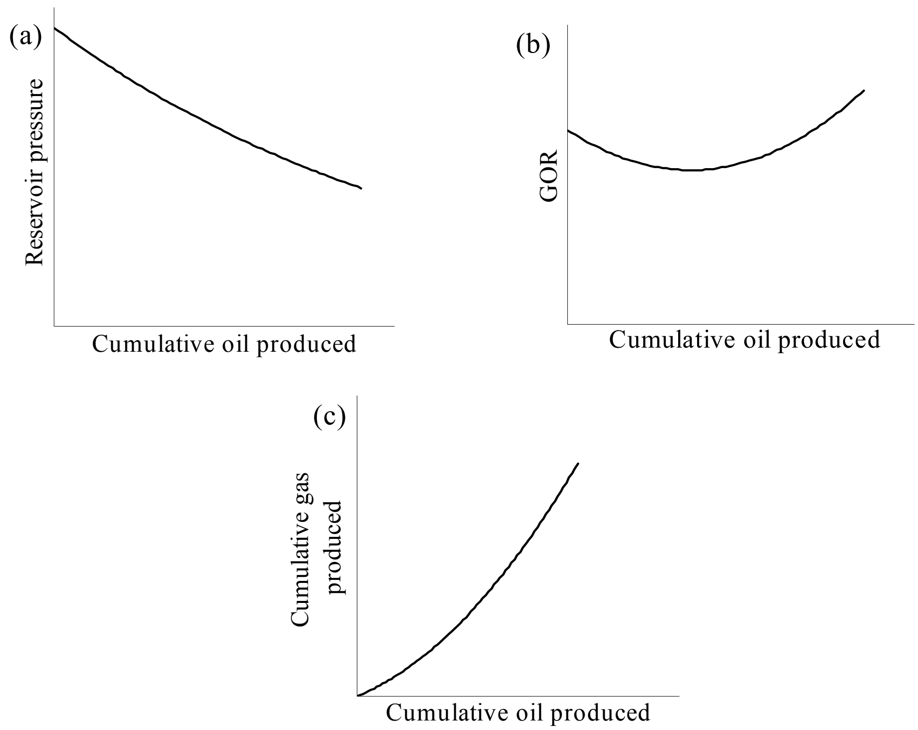


**Figure 2:** An example of (a) reservoir pressure as a function of the cumulative oil produced, (b) gas-to-oil ratio (GOR) as a function of cumulative oil produced and (c) cumulative gas produced as a function of cumulative oil produced

## Solution Strategy

Even though the elimination of linear interpolation binary variables reduces the problem size significantly, the resulting model (P) still becomes very large as the number of time periods increases. This is due to the increase of constraints and variables with each additional period. In addition to this difficulty, the newly included nonlinear constraints introduce non-convexities into the model, thereby increasing the chance of finding sub-optimal solutions. A specialized solution strategy capable of dealing with both the problem size and the non-convexities is therefore needed. We propose such a strategy based on logic-based methods, the use of convex envelopes, bilevel decomposition, and the aggregation of time periods. In the following four sections we discuss these aspects and show how we integrate them to form an iterative aggregation/disaggregation algorithm for oilfield infrastructure planning.

## Logic-Based Methods

Model (P) is logic-based since it is in disjunctive form, and a logic-based method is required for the solution thereof. Turkay and Grossmann (1996a) proposed a logic-based Outer Approximation (OA) algorithm for MINLPs based on the OA method by Duran and Grossmann (1986). The latter 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 method, linearizations are only added to the MILP if the disjunction has true value, and the NLP subproblems only include constraints for existing units (i.e. constraints of disjunctions with true value). Both problems are converted to mixed-integer form through the convex hull formulation (Balas, 1985; see also Turkay and Grossmann). The advantage of this formulation is that it reduces the dimensionality of the nonlinear sub-problem by only considering disjunctions for which the Boolean variable is true, thereby avoiding singularities due to linearizations at zero flows, and eliminating non-convexities of non-existing processes. Van den Heever and Grossmann (1999) show that this algorithm can successfully be applied to multiperiod problems.

In this paper, we extend the logic-based OA algorithm to deal with non-convexities, by using convex envelopes where non-convexities occur, instead of linearizations, as discussed in the next paragraph.

## Dealing with non-convexities

When non-convexities are present in an optimization model, the OA algorithm cannot guarantee a global optimum. Standard NLP solvers such as MINOS and CONOPT cannot guarantee a global optimum for the NLP sub-problem, and the NLP solution often depends on the starting point given by the modeler. For the MILP master problems, parts of the feasible region may be cut off by linearizations of non-convex constraints (see Figure 3). When this happens, it is possible that the optimal solution is eliminated from the optimization procedure and never found. The crucial drawback therefore lies in the MILP, since a large number of feasible solutions, including the optimal solution, might be missed completely. We address this problem by replacing the linearizations of the non-convex nonlinear functions in model (P) with convex envelopes (Horst, 1995).

Consider, for example, the quadratic equation representing cumulative gas as a function of cumulative oil as shown in Figure 3(a). In the MILP step of the OA algorithm, this constraint is relaxed into two inequalities, one greater or equal, and one less or equal, as shown by the solid lines in Figures 3(b) and 3(c). The vertical arrows indicate which part of the feasible region is subsequently cut off. The next step in the OA algorithm is to linearize the inequalities at a point obtained from the NLP sub-problem, as shown by the dotted lines in Figures 3(b) and 3(c). For the case of a convex inequality, shown in Figure 3(b), the linearization relaxes the feasible region further. However, in the case of the non-convex inequality (Figure 3(c)), the linearization cuts off a part of the feasible region as shown by the horizontal lines. To overcome this problem, we make use of a convex envelope where we replace the linearization of the non-convex inequality with a straight line between the upper and lower bounds as shown in Figure 3(d).

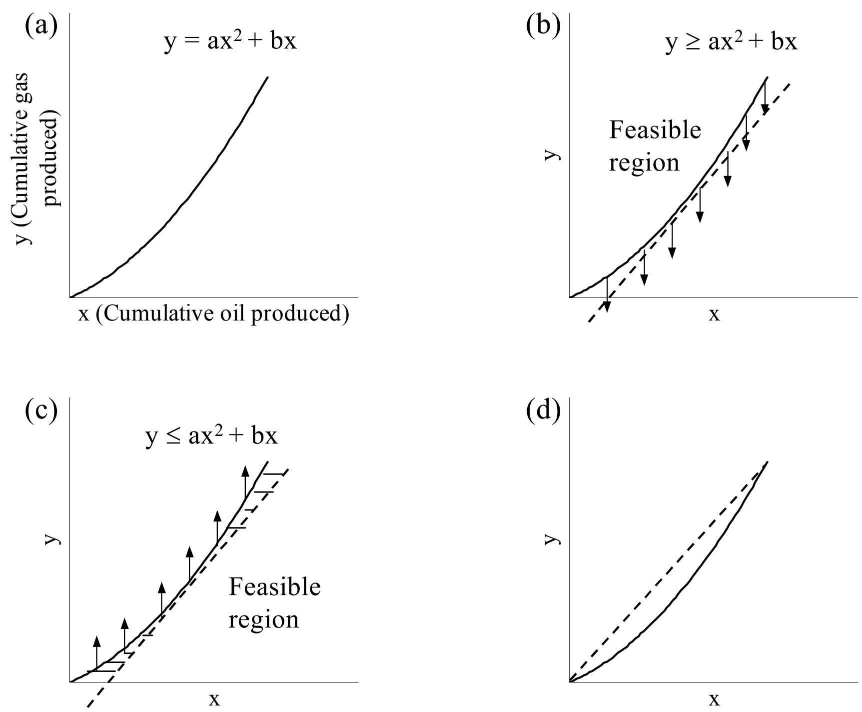


**Figure 3:** (a) Cumulative gas produced as a quadratic function of cumulative oil produced, (b) convex inequality relaxation with linearization, (c) linearization of non-convex inequality cuts off part of the feasible region, (d) convex envelope for the quadratic function

In general, for any nonlinear equality of the form $x^i = g(x^j)$, where $x^i, x^j \in x_t$, the equality is relaxed as two inequalities, $x^i \leq g(x^j)$ and $x^i \geq g(x^j)$. If $g(x^j)$ is a convex function, then $- x^i + g(x^j) \leq 0$ is convex and $x^i - g(x^j) \leq 0$ is concave. For the convex case, linearization at a fixed value $x^{j,k}$ yields a valid relaxation:

$$x^i \geq g(x^{j,k}) + \nabla_{x^j} g(x^{j,k})(x^j - x^{j,k}) \quad \text{(a)}$$

Here the index $k$ indicates that the fixed value of $x^j$ is obtained from the $k^{th}$ NLP in the OA algorithm. The linearization set for active terms of the disjunctions, given $L$ major iterations, are defined as $K = \{k \mid z_t = True, k = 1..L, t = 1..T\}$.

For the concave case we need to use a convex envelope for a valid relaxation over the domain $x^j \in [x^j_L, x^j_U]$, where $L$ and $U$ indicate the upper and lower bounds respectively:

$$x^i \leq g(x^j_L) - \left( \frac{g(x^j_U) - g(x^j_L)}{x^j_U - x^j_L} \right) \left( x^j - x^j_L \right) \qquad \text{(b)}$$

The multiplier of $x^i = g(x^j)$ from the NLP subproblem indicates in which way the equality relaxes, and determines which of (a) or (b) will be used in the MILP master problem. For simplicity, we write (a) as $f_t^{lin,k}(x_t) \leq 0$ and (b) as $f_t^{env}(x_t) \leq 0$ from now on. The master problem of the extended OA algorithm can then be written as:

$$\max \Psi(x, z)$$

subject to

$$\begin{aligned}
&Ax_t \leq b && \forall t = 1..T \\
&f_t^{lin,k}(x_t) \leq 0 && \forall t = 1..T \\
&f_t^{env}(x_t) \leq 0 && \forall t = 1..T \\
&g_t(x_1, x_2..x_t) \leq 0 && \forall t = 1..T
\end{aligned}$$

$$
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_\theta^{p} \\
A_p x_t^{p} \le b_p \\
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_\theta^{\pi,p} \\
A_\pi x_t^{\pi,p} \le b_\pi \\
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_\theta^{w,\pi,p} \\
A_w x_t^{w,\pi,p} \le b_w \\
h_{t,w}^{lin,k}(x_t^{w,\pi,p}) \le 0 \qquad \forall k \in K \\
h_{t,w}^{env}(x_t^{w,\pi,p}) \le 0 \\
g_{t,w}(x_1^{w,\pi,p}, x_2^{w,\pi,p} .. x_t^{w,\pi,p}) \le 0
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_\theta^{w,\pi,p} \\
B^{w,\pi,p,t} x_t^{w,\pi,p} = 0
\end{array}\right] \\
\forall w \in W_{WP}(\pi)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_\theta^{\pi,p} \\
B^{\pi,p,t} x_t^{\pi,p} = 0
\end{array}\right] \\
\forall \pi \in WP(p)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_\theta^{p} \\
B^{p,t} x_t^{p} = 0
\end{array}\right] \qquad \text{(MP)}
$$

$$
\forall p \in PP, t \in T
$$

$$
\Omega(z) = True
$$

$$
x \ge 0, \quad z \in \{True, False\}
$$

The disjunctive master problem (MP) is converted into mixed integer form (model (MIP)) by obtaining the convex hull of each disjunction as shown in Appendix B. The disjunctive NLP sub-problem is given by the following formulation where $\bar{z}$ indicates a fixed value of the Boolean variable as obtained from the MILP master problem:

$$
\max \Psi(x, \bar{z})
$$

subject to

$$
\begin{array}{ll}
Ax_t \le b & \forall t = 1..T \\
f_t(x_t) \le 0 & \forall t = 1..T \\
g_t(x_1, x_2..x_t) \le 0 & \forall t = 1..T
\end{array}
$$

$$
\left.\begin{array}{l}
\left.\begin{array}{l}
\left.\begin{array}{l}
A_w x_t^{w,\pi,p} \le b_w \\
h_{t,w}(x_t^{w,\pi,p}) \le 0 \\
g_{t,w}(x_1^{w,\pi,p}, x_2^{w,\pi,p}..x_t^{w,\pi,p}) \le 0
\end{array}\right\} \bigvee_{\theta=1}^{t} \bar{z}_\theta^{w,\pi,p} = True \\
B^{w,\pi,p,t} x_t^{w,\pi,p} = 0 \qquad \left.\right\} \bigvee_{\theta=1}^{t} \bar{z}_\theta^{w,\pi,p} = False \\
A_\pi x_t^{\pi,p} \le b_\pi
\end{array}\right\} \bigvee_{\theta=1}^{t} \bar{z}_\theta^{\pi,p} = True \\
B^{\pi,p,t} x_t^{\pi,p} = 0 \qquad \left.\right\} \bigvee_{\theta=1}^{t} \bar{z}_\theta^{\pi,p} = False \\
A_p x_t^p \le b_p
\end{array}\right\} \bigvee_{\theta=1}^{t} \bar{z}_\theta^{p} = True
$$

$$
B^{p,t} x_t^p = 0 \qquad \left.\right\} \bigvee_{\theta=1}^{t} \bar{z}_\theta^{p} = False \qquad \text{(NP)}
$$

$$
x \ge 0
$$

Figure 4 shows the logic-based OA algorithm as integrated in this work.

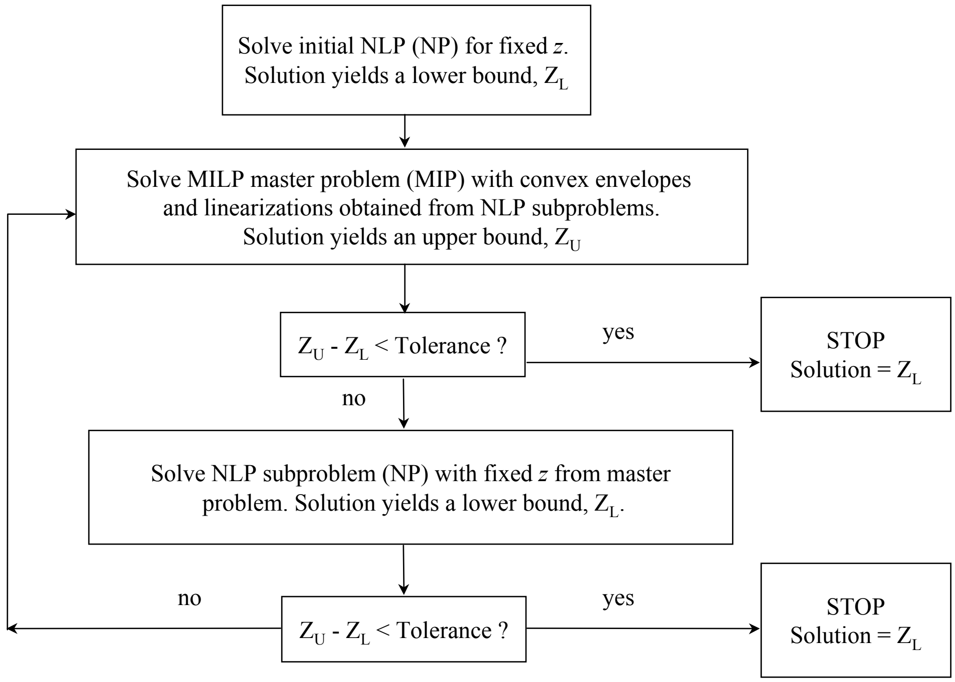


**Figure 4:** Logic-based OA algorithm

**Example 1**

To illustrate the benefit of the use of convex envelopes, we solve a small instance of the oilfield infrastructure planning problem consisting of 1 PP, 1 WP, and 2 reservoirs with a total of 6 wells. This problem is solved for the complete planning horizon of 6 years, divided into 24 quarterly periods. The disjuntive model, when formulated as an MINLP, contains 2803 constraints, 1729 continuous variables and 225 binary variables. These binary variables are treated as SOS1 variables in all the examples. The data is the same data as that used by Iyer *et al*., but we expect a different objective value from the one obtained by Iyer *et al*., since we do not consider the cost involved in moving drilling rigs. Table 2 shows the comparison of objective value for different starting points when we solve this example without the use of convex envelopes versus when we include the convex envelopes.

**Table 2:** Comparison of objective value with and without use of convex envelopes

| | Starting point | Objective ($mil.) |
|---|---|---|
| OA with linearizations only | Relaxed | 5.24 |
| | NLP | 8.38 |
| OA with linearizations and convex envelopes | Relaxed | 8.46 |
| | NLP | 8.46 |

To find the solution, we programmed the OA algorithm, as used in DICOPT++, manually in GAMS in order to modify the algorithm to include convex envelopes and to allow the use of different starting points. When the starting point is the relaxed MINLP solution and convex envelopes are included in the model, we find the objective value of $ 8.46 million. In this solution one well was chosen from each reservoir to be drilled in the first time period. However, when convex envelopes are not included, a sub-optimal solution of $5.24 million, where one well from each reservoir is drilled in the 1st period and three wells are drilled in the 5th period, is obtained. By using a different starting point, namely a feasible NLP solution, the objective without convex envelopes improves to a value of $8.38 million, which is again sub-optimal compared to the solution when convex envelopes are included.

**Bilevel Decomposition**

The bilevel decomposition approach, first introduced by Iyer and Grossmann (1998), reduces the problem size by decomposing the original design and planning model into an upper level design problem and a lower level planning problem. The upper level design problem (DP) includes only the discrete variables representing design decisions, in our context whether or not to include a well or platform during the whole planning horizon. The lower level planning problem (OP) is solved for a fixed design or configuration and determines the operational plan in each time period, for example when to drill a well. (DP) is a relaxation of the original model, since only design decisions are considered, and thus yields an upper bound. (OP) is in a reduced solution space, since a subset of fixed discrete variables are used as obtained from (DP), making it possible to ignore a large number of equations and variables. A lower bound is obtained from (OP) since its solution corresponds to a feasible solution of the original model. Variable values from the previous (OP) solution are used to formulate cuts for problem (DP). Integer cuts are used to exclude subsets and supersets of previously obtained feasible configurations and to exclude infeasible configurations from future calculations. Design cuts are used to force values of state variables in (DP) to be greater than or equal to their values in (OP) if the same configuration is chosen in both problems. The solution of (OP) with the current upper bound is the final solution after convergence is achieved.

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. Van den Heever and Grossmann (1999) extended the method to be applicable to nonlinear problems, by formulating the optimization model as a disjunctive problem and solving the sub-problems with the logic-based OA algorithm.

## Aggregation of Time Periods

Due to the size of the oilfield infrastructure planning problem, it cannot be solved in reasonable time with the disjunctive bilevel decomposition algorithm. To address this difficulty, we propose the aggregation of time periods. Rogers et al. (1991) give a thorough review of aggregation and disaggregation techniques and methodology in optimization. Specific applications in the chemical industry include a rigorous general aggregation methodology for large-scale process scheduling problems (Wilkinson,1996), and the method used by Iyer *et al*. (1998). Both these methods first solve a time aggregate subproblem (see Figure 5), and then each aggregated time period is disaggregated and solved sequentially to find a feasible solution.

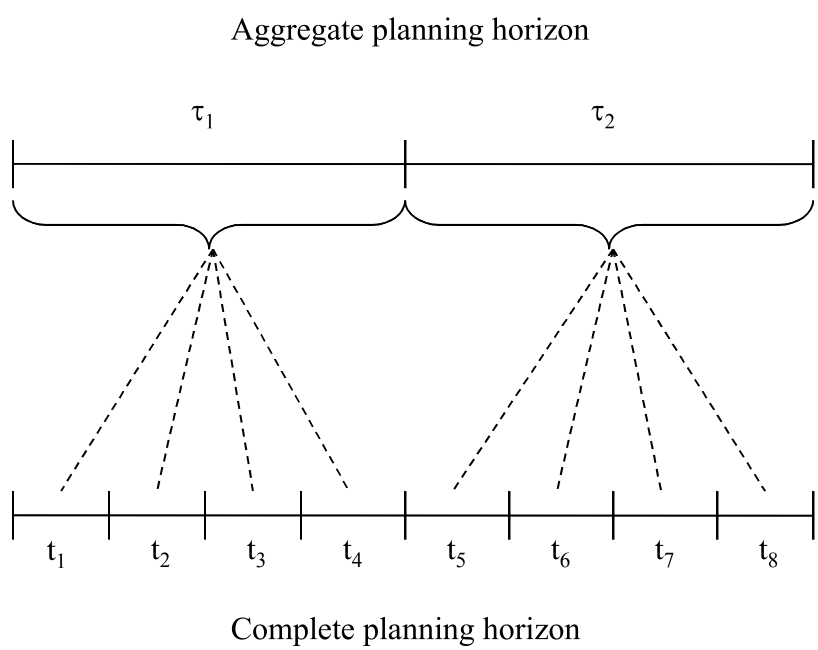


**Figure 5:** Aggregation of time periods

The methods mentioned above perform the aggregation/disaggregation procedure once only in order to obtain *a* feasible solution. We propose an iterative scheme (see Figure 6) in order to try and improve on the first feasible solution by using information obtained from previous iterations. For this purpose, we integrate the aggregation of time periods with the logic-based bilevel decomposition algorithm proposed by Van den Heever and Grossmann (1999), while incorporating the use of convex envelopes. When performing the aggregation, we consider the hierarchy of the model, namely that design decisions are on a higher hierarchical level than planning decisions. The design problem is therefore aggregated in time and solved to yield an upper bound and a fixed infrastructure. Subsequently, the time periods are disaggregated and the planning problem is solved for the fixed infrastructure to yield a lower bound. The aggregate design problem (ADP) is as follows:

$$\max \Psi(x,z)$$

subject to

$$\begin{aligned} & Ax_\tau \le b && \forall \tau = 1..TA \\ & f_\tau(x_\tau) \le 0 && \forall \tau = 1..TA \\ & g_\tau(x_1, x_2 .. x_\tau) \le 0 && \forall \tau = 1..TA \end{aligned}$$

$$\begin{bmatrix} \bigvee_{\theta=1}^{\tau} z_\theta^p \\ A_p x_\tau^p \le b_p \\ \begin{bmatrix} \bigvee_{\theta=1}^{\tau} z_\theta^{\pi,p} \\ A_\pi x_\tau^{\pi,p} \le b_\pi \\ \begin{bmatrix} \bigvee_{\theta=1}^{\tau} z_\theta^{w,\pi,p} \\ A_w x_\tau^{w,\pi,p} \le b_w \\ h_{\tau,w}(x_\tau^{w,\pi,p}) \le 0 \\ g_{\tau,w}(x_1^{w,\pi,p}, x_2^{w,\pi,p} .. x_\tau^{w,\pi,p}) \le 0 \end{bmatrix} \vee \begin{bmatrix} \neg \bigvee_{\theta=1}^{\tau} z_\theta^{w,\pi,p} \\ B^{w,\pi,p,\tau} x_\tau^{w,\pi,p} = 0 \end{bmatrix} \\ \forall w \in W_{WP}(\pi) \end{bmatrix} \vee \begin{bmatrix} \neg \bigvee_{\theta=1}^{\tau} z_\theta^{\pi,p} \\ B^{\pi,p,\tau} x_\tau^{\pi,p} = 0 \end{bmatrix} \\ \forall \pi \in WP(p) \end{bmatrix} \vee \begin{bmatrix} \neg \bigvee_{\theta=1}^{\tau} z_\theta^{p} \\ B^{p,\tau} x_\tau^{p} = 0 \end{bmatrix}$$

$$\forall p \in PP, \tau \in TA \qquad \text{(ADP)}$$

$$\Omega(z) = True$$

$$x \ge 0, \quad z \in \{True, False\}$$

Constraints are now valid for each aggregate period τ ∈ TA. The duration of time periods that appear in the constraints of the original model given in Appendix A, are replaced by the duration of the aggregate time periods. In order to insure that an upper bound results from the aggregate design problem, the discounting factors in the objective function need to be chosen in such a way as to overestimate the objective and the production also needs to be overestimated. Iyer *et al*. present a proof that this can be accomplished by specifying that the flow profile is non-increasing for each well, and that investments are discounted as if they occur at the end of an aggregate period, while sales and depreciation are discounted as if they occur at the beginning of an aggregate period. We incorporate this property in our aggregate model to insure an upper bound.

The disaggregate planning problem has the same formulation as model (P), except that it is solved for a subset of wells and platforms as obtained from (ADP). Also, the solution form (ADP) is used to curb the investment horizon for each well in the planning problem. This is done by specifying that well *w* can only be drilled in disaggregate period *t* if $\bigvee_{\theta=1}^{\tau} z_{\theta}^{w,\pi,p} = True$ and $t \leq \sum_{\theta=1}^{\tau} m_{\tau}$ , where $m_{\tau}$ is the length of aggregate period $\tau$.

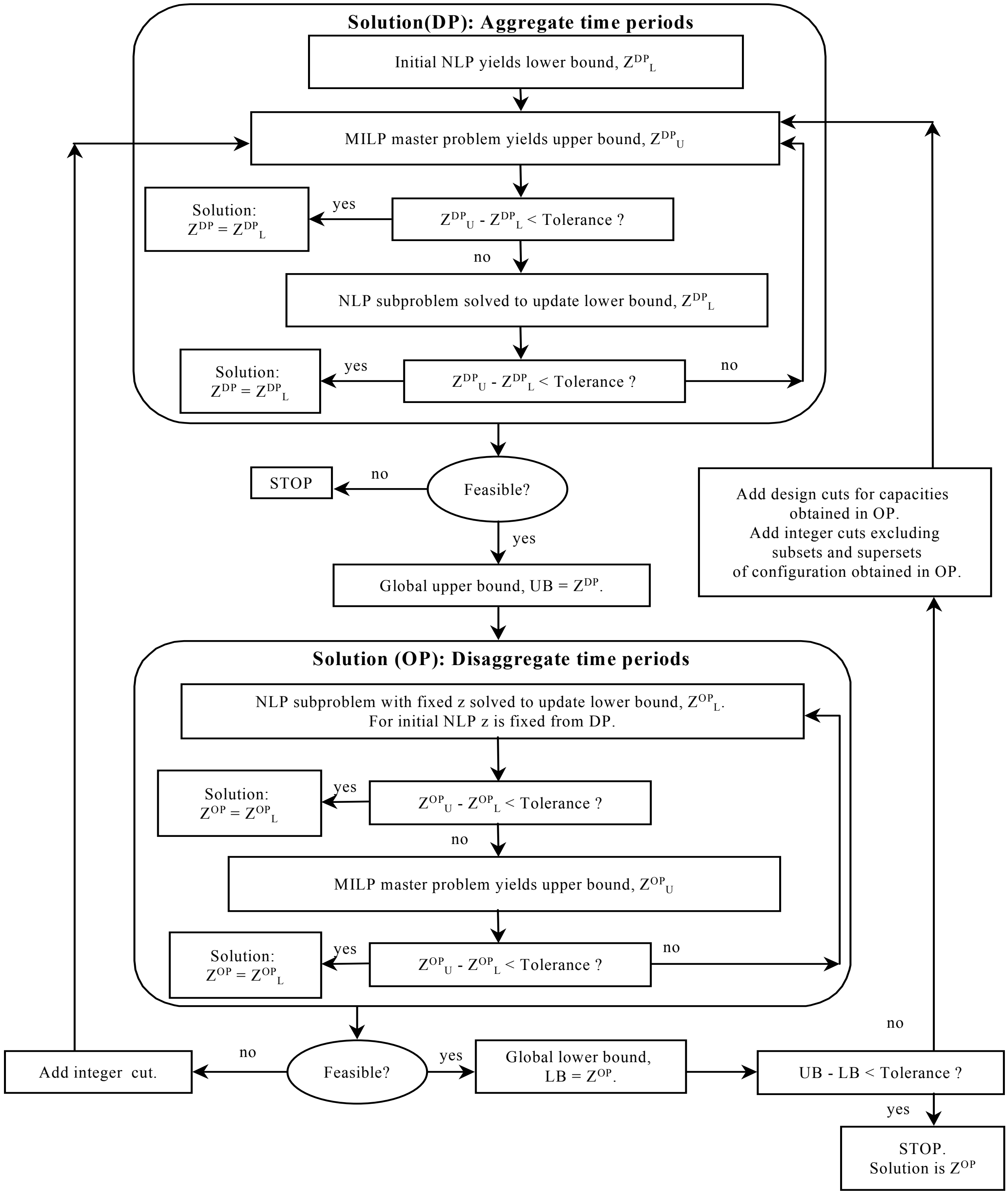


**Figure 6:** Disjunctive bilevel decomposition algorithm with aggregation of time periods

**Example 1 (continued)**

The iterative aggregation/disaggregation algorithm, as shown in Figure 6, is applied to Example 1 in which we use 6 aggregate time periods to represent the 24 original time periods. This means that the MINLP problem size is reduced from 2803 constraints, 1729 continuous variables and 225 binary variables, to 802 constraints, 432 continuous variables and 63 binary variables. The solution time is compared to the solution time when we solve the same problem with the OA algorithm (see Table 3). In this version of the OA algorithm, we incorporate convex envelopes into the model in order to obtain a reasonable objective value for comparison. Both algorithms are coded in the GAMS modeling language and solved on a HP9000/C110 workstation. All NLP sub-problems are solved with CONOPT2, while all MILP master problems are solved with CPLEX 6.5.

**Table 3:** Results for example 1 – comparison of solution times

| | Best LB ($mil.) | Solution time (CPU sec.) |
|---|---|---|
| **OA** | 8.46 | 538 |
| **Proposed method** | 8.46 | 37 |

The same objective value is obtained by both algorithms after 3 iterations. While it took the OA algorithm 538 CPU seconds to find this solution, the proposed method took only 37 CPU seconds. This is an order of magnitude improvement in solution time.

However, the improvement in solution time is offset by the large gap between the aggregate and disaggegate subproblem solutions. For this instance, the aggregate subproblem (ADP) yields an upper bound of $ 10.36 million giving a gap of 22.6 % between upper and lower bound. A gap between the solution of (ADP) and (OP) will always exist due to the aggregation, but the drawback of too large a gap is that the bounds do not give a good indication of how close the lower bound is to the actual optimal solution.

The difficulty that arises here is that the aggregate problem gives poor upper bounds depending on the aggregation scheme, i.e. which time periods are grouped together. This difficulty can be seen in Figure 7 for eight time periods. In Figure 7(a), the time periods have been bundled together in groups of four and the graph shows the distribution of income versus aggregate time. When the time periods are subsequently disaggregated, the income distribution in Figure 7(b) is found. It is clear from Figure 7(c) that if the time periods were aggregated in one group of two and one group of six instead, the aggregate problem would resemble the disaggregate one of Figure 7(b) more closely.

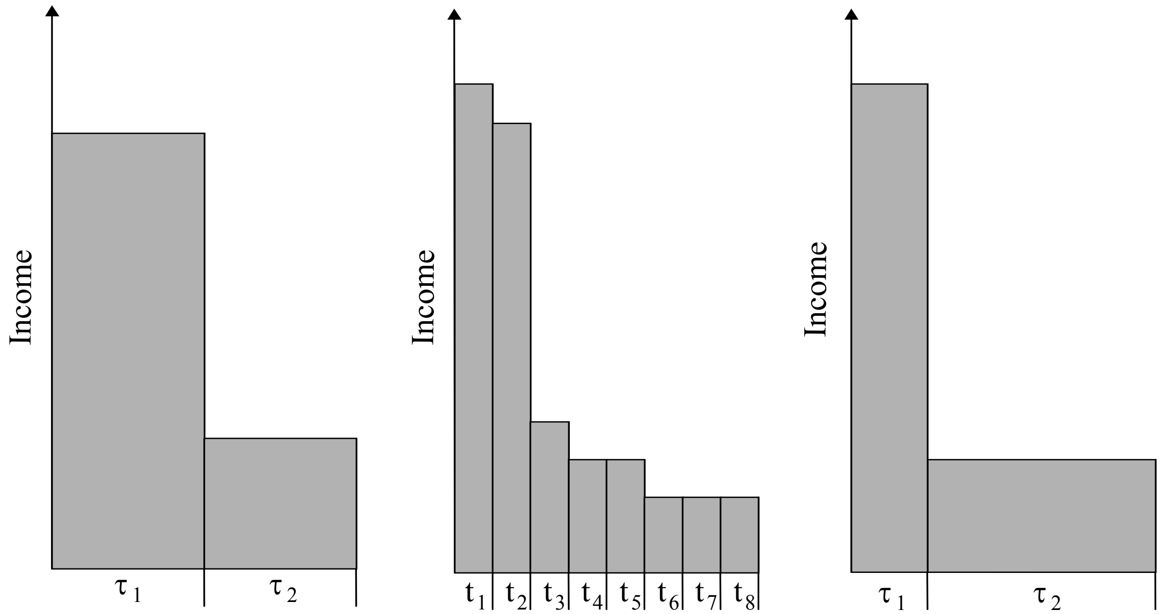


**Figure 7:** (a) Income distribution for initial aggregation scheme grouping time periods into groups of four; (b) Income distribution from disaggregate solution; (c) Aggregation into one group of two and one group of six resembles disaggregate income distribution better.

However, this cannot be known beforehand and by finding a poor upper bound, we might unneccesarily dismiss a good solution. The question is therefore, how should the time periods be bundled together so that the aggregate problem resembles the disaggregate problem as close as possible?

**Aggregation Improvement Subproblem (PAI)**

We propose an intermediate optimization problem where the basic idea is to determine the lengths of the aggregate time periods that minimizing the NPV, subject to the constraint that the NPV is an upper bound to the original problem. We formulate and solve this intermediate problem (PIA) with a dynamic programming approach (Bellman, 1957) as follows:

Sets:

T disaggregate time periods

TA aggregate time periods

Indices:

t disaggregate time period t $\in$ T

$\tau$ aggregate time period $\tau \in$ TA

Parameters:

$f_{inv,t}$ discounting factor for investment in period t

$f_{dpr,t}$ discounting factor for depreciation in period t

$f_{rev,t}$ discounting factor for revenue in period t

$I_t$ investment costs in period t

$R_t$ revenue in period t

Variables:

$s_\tau$ state at the end of aggregate period $\tau$, i.e. number of disaggregate periods available for assignment at the end of aggregate period $\tau$

$m_\tau$ length of aggregate period $\tau$

Recursive scheme:

$$h_\tau(s_{\tau-1}) = \min_{m_\tau}\left[g_\tau(s_{\tau-1}, m_\tau) + h_{\tau+1}(s_\tau)\right] \qquad \forall \tau = TA..1 \qquad (1)$$

$$s_\tau = s_{\tau-1} - m_\tau \qquad \forall \tau = TA..1 \qquad (2)$$

$$g_\tau(s_{\tau-1}, m_\tau) = \left[ f_{inv,(T-s_{\tau-1}+m_\tau)}(1 - f_{dpr,(T-s_{\tau-1}+1)}) \sum_{t=T-s_{\tau-1}+1}^{t=T-s_{\tau-1}+m_\tau} I_t \right] + f_{rev,(T-s_{\tau-1}+1)} \sum_{t=T-s_{\tau-1}+1}^{t=T-s_{\tau-1}+m_\tau} R_t \qquad \text{(PAI)}$$

$$\forall \tau = TA..1 \qquad (3)$$

$$h_{TA+1}(s_{TA}) = 0 \qquad (4)$$

The objective function (1) is to minimize the NPV. This is counter-intuitive, since one would normally expect to maximize NPV. However, by chosing the discounting factors appropriately, we ensure that the optimal value is an upper bound to the original disaggregate problem. The optimization has the effect of determining the length of aggregate time periods such that we obtain the lowest possible upper bound, i.e. the upper bound that is closest to the previous lower bound. The investments, $I_t$, and revenues, $R_t$, in each time period are obtained from the solution of the previous disaggregate problem. The objective function is calculated recursively for each stage/aggregate time period from *TA* down to 1. Constraint (2) relates the current state to the previous state through the length of the current aggregate time period. The number of disaggregate periods available for assignment to remaining aggregate periods equals the number of periods that were available at the previous stage minus the number that are assigned to the current aggregate period. Constraint (3) calculates the actual objective value at each stage of recursion for all possible values of the state at that stage, while constraint (4) gives the initial condition for the start of recursion. The final solution is given by $h_1(s_0)$. We programmed this model in C++ and were able to solve it in less than 1 CPU second for all instances. An illustrative example showing input and solution format for this model is given in Appendix C.

**Example 2**

The solution of (PAI) yields an objective value which is an indication of how close the upper bound can get to the lower bound, as well as an aggregation scheme to obtain an improved upper bound. The short solution time allows the possibility of experimenting with different numbers of total aggregate periods to determine if we can have an even smaller aggregate problem without losing too much accuracy. To illustrate this we solve (PAI) for different numbers of aggregate periods with data obtained from the solution of (OP) for the problem described in Example 1. The results (see Table 4) show that as the number of aggregate periods increase, the quality of the best possible bound deteriorates. This is expected since accuracy worsens when constraints are aggregated. According to Table 4, by letting TA = 6, an upper bound no less than $ 8.98 million should be obtained if the time periods are aggregated according to the scheme in the rightmost column. For our purposes, we choose to fix the number of aggregate periods to 6, since this allows a significant speed-up in solution time, while keeping an appropriate level of accuracy (an expected gap of 6.1% between bounds).

**Table 4:** Quality of the upper bound (from PAI) versus number of aggregate time periods

| TA (# aggregate time periods) | Objective ($ mil.) | Aggregation scheme |
|---|---|---|
| 3 | 10.11 | 1 ,7 ,16 |
| 6 | 8.98 | 1, 2, 3, 3, 3, 12 |
| 12 | 8.56 | 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 10 |
| 24 | 8.46 | 1 each |

## Iterative Aggregation/Disaggregation Algorithm

As a final step, model (PIA) is integrated into the algorithm presented in Figure 5, to yield the final iterative aggregation/disaggregation algorithm in Figure 8. After the first iteration, the solution from (OP) is used as input parameters for (PAI), which is subsequently solved to obtain a new aggregation scheme, as well as an indication of how good the upper bound from (DP) can be expected to be. (DP) is then aggregated with the new scheme and solved to obtain a new upper bound. The algorithm is terminated if there has been no improvement in two iterations. A feasible NLP solution is used as the starting point for all sub-problems.

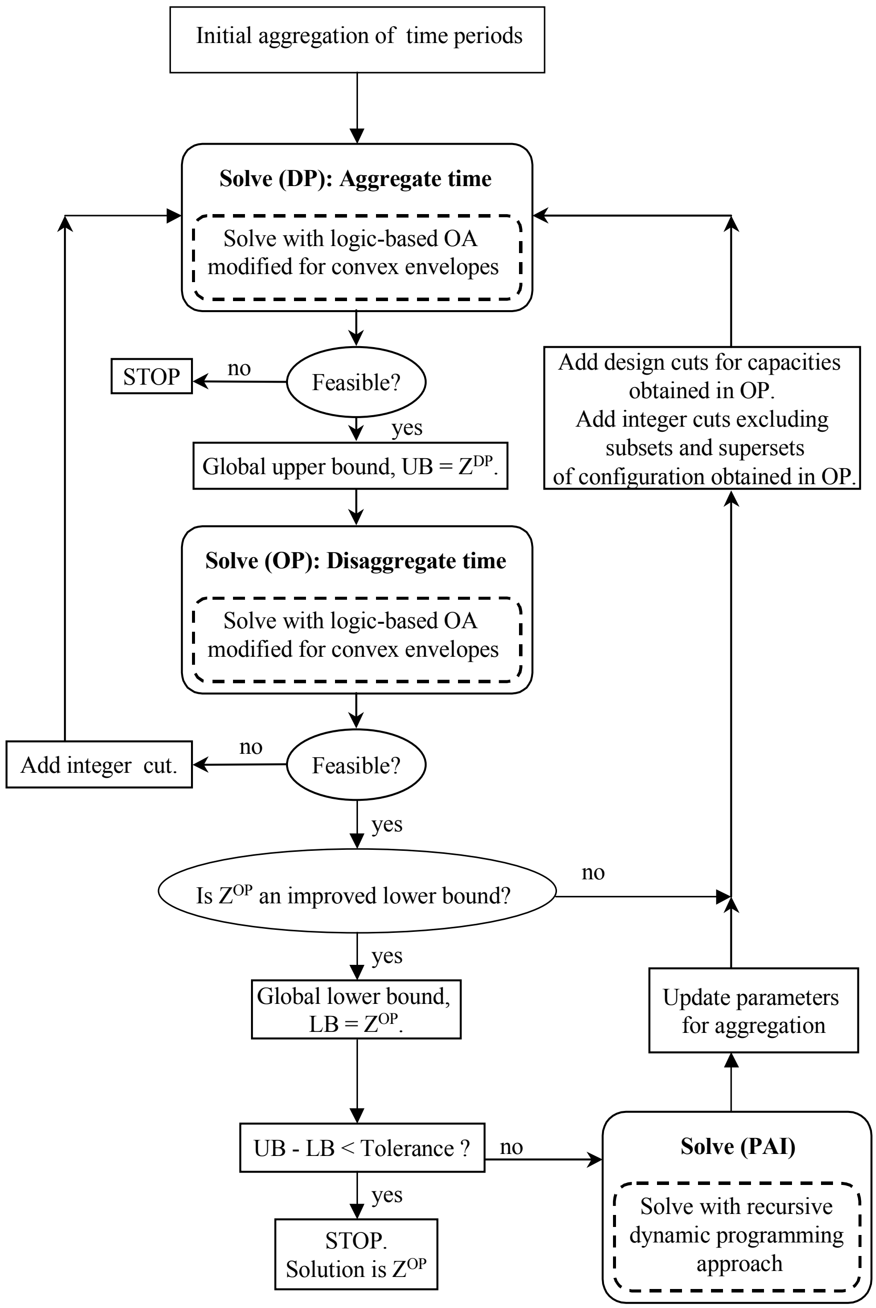


**Figure 8:** Iterative aggregation/disaggregation algorithm

**Example 1 (continued)**

Example 1 is solved with the final proposed algorithm shown in Figure 8, as well as with the initial algorithm proposed in Figure 6, and the gap between the bounds are compared. Table 5 shows that when sub-problem (PAI) is not included in the algorithm for an aggregation into 6 aggregate periods of length 4 each, we obtain an upper bound of $ 10.37 million, with a gap of 22.6 %. If one were to judge the solution, that is the current lower bound of $ 8.46 million, one might ignore it and keep searching for a better one since it seemingly can reach a value of $10.37 million. However, the incorporation of (PAI) into the algorithm leads to an improved aggregation scheme for 6 aggregate periods of lengths (1,2,3,3,3,12) by which the more accurate upper bound of $ 8.98 million is obtained, with a gap of 6.1 %. This gap is the same as the one predicted by (PAI). The algorithm is terminated when no improvement is observed in two iterations after the previous best solution.

**Table 5:** Comparison of upper bound obtained

| | Lower bound ($mil.) | Upper bound ($ mil.) | Gap (%) |
|---|---|---|---|
| **Without (PAI)** | 8.46 | 10.37 | 22.6 |
| **With (PAI)** | 8.46 | 8.98 | 6.1 |

**Example 3**

In this section we solve the oilfield infrastructure planning problem for a range of problem sizes up to the largest instance consisting of 1 PP, 2 WPs, and 10 reservoirs containing a total of 25 well sites. All problems are solved for the complete horizon of 24 time periods and the results are presented in Table 6. The first four columns show the various MINLP problem sizes in terms of number of constraints, total number of variables, and number of binary variables (treated as SOS1 variables) for the given number of wells. The next two columns show a comparison of the best lower bounds obtained by the OA algorithm (as used in DICOPT++) and the proposed algorithm respectively. Note that the OA algorithm is coded in GAMS to include convex envelopes in order to obtain reasonable lower bounds. From these values it can be seen that the bounds obtained by the proposed method are very similar to, or slightly better than, the ones obtained when the problems are solved in the full space by the OA algorithm. This indicates that accuracy is maintained in the proposed decomposition and aggregation/disaggregation.

The next two colums ($7^{th}$ and $8^{th}$) show the solution times in CPU seconds, obtained on a HP9000/C110 workstation, for the fullspace OA algorithm and the proposed algorithm respectively. All NLP sub-problems are solved with CONOPT2, while all MILP master problems are solved with CPLEX 6.5. The results show an order of magnitude reduction in solution time for the proposed method compared to the OA algorithm as used in DICOPT++. For the largest instance the OA algorithm solved in 19385.9 CPU seconds, while the proposed method solved the problem in 1423.4 CPU seconds.

The last two columns compare the gap between the upper bound obtained from the aggregate design problem and the lower bound obtained from the disaggregate planning problem for the cases when sub-problem (PAI) is and is not included in the algorithm. For the smallest problem instance (4 wells) a 77% reduction in the gap (from 21.4% to 5.0%) is obtained when (PAI) is included in the algorithm. The reduction in the gap decreases with growth in problem size, down to a 31.0% reduction for the largest instance (from 12.6% to 8.7%). This can be explained by noticing that more accuracy is lost when a problem with a large number of wells is aggregated, than when a problem with a small number of wells is aggregated. The gap when (PAI) is included ranges between 3.7% and 8.7%, while the gap when (PAI) is not included ranges between 8.5% and 22.6%. A better indication of the quality of the solution is therefore obtained by the inclusion of (PAI) into the algorithm.

**Table 6:** Results for Example 3

| **Problem size (24 time periods)** | | | | **Best LB* ($mil.)** | | **Solution time* (CPU sec.)** | | **Gap w/o (PAI) (%)** | **Gap w/ (PAI) (%)** |
|---|---|---|---|---|---|---|---|---|---|
| **# wells** | **# constraints** | **# variables** | **# 0-1 variables** | **GAMS (OA)** | **Proposed method** | **GAMS (OA)** | **Proposed method** | | |
| 4 | 1917 | 1351 | 150 | 8.55 | 8.55 | 224.3 | 31.1 | 21.4 | 5.0 |
| 6 | 2803 | 1954 | 225 | 8.46 | 8.46 | 538.0 | 37.0 | 22.6 | 6.1 |
| 9 | 3880 | 2677 | 300 | 25.86 | 25.86 | 3936.7 | 113.8 | 10.7 | 3.9 |
| 11 | 4622 | 3183 | 350 | 36.48 | 36.44 | 2744.2 | 227.1 | 9.1 | 3.7 |
| 14 | 5699 | 3906 | 425 | 49.40 | 49.40 | 3804.8 | 434.3 | 8.5 | 4.0 |
| 16 | 6441 | 4412 | 475 | 54.25 | 54.55 | 4944.8 | 809.0 | 11.6 | 7.0 |
| 19 | 7518 | 5135 | 550 | 64.08 | 64.35 | 6521.2 | 887.6 | 9.8 | 6.0 |
| 21 | 8260 | 5641 | 600 | 63.86 | 64.33 | 11659.3 | 1163.1 | 9.7 | 6.3 |
| 23 | 9002 | 6147 | 650 | *** | 65.62 | *** | 1211.6 | 9.8 | 6.0 |
| 25 | 9744 | 6653 | 700 | 67.90 | 67.99 | 19385.9 | 1423.4 | 12.6 | 8.7 |

Figures 9 and 10 show the final solution obtained for the largest problem instance of 25 wells for the planning horizon of 24 periods. The final configuration is shown in Figure 9. Note that only 9 of the potential 25 wells are drilled over the 24 periods. Of these, 3 are drilled in the first period, 1 in the second, 1 in the third, 1 in the fourth, and 3 in the fifth period.

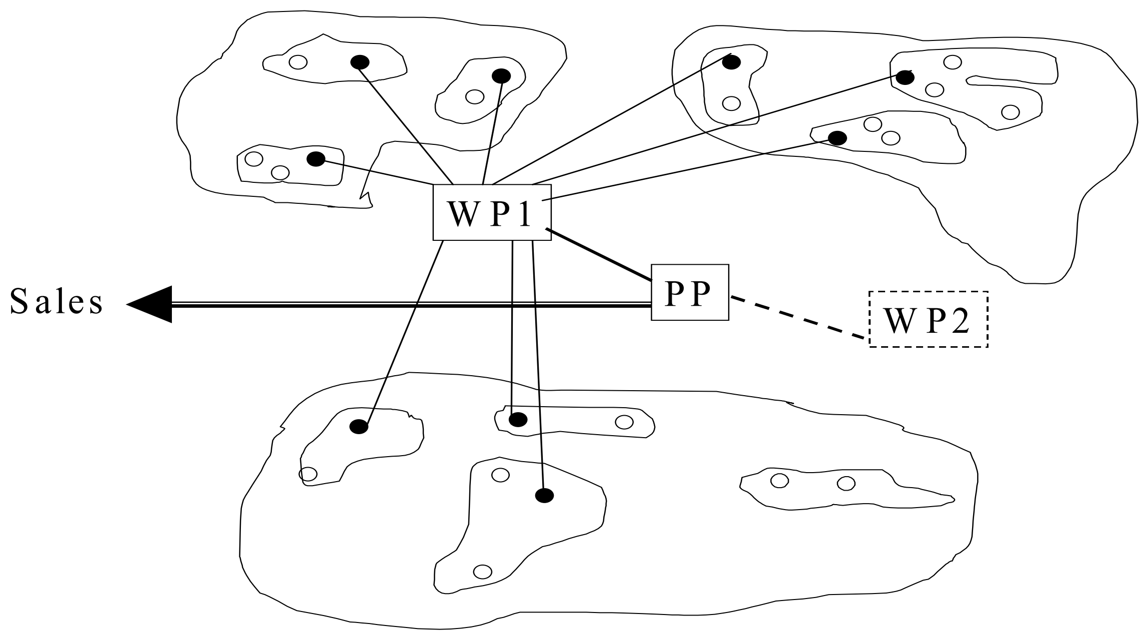


**Figure 9:** The final configuration.

Figure 10 shows the production profile for the whole infrastructure over the 24 time periods encompassing the six years from January 1999 up to December 2004. A profit of $ 67.99 million is obtained. This final solution is found in less than 25 minutes by the proposed algorithm, whereas a traditional solution approach such as the OA algorithm needs more than 5 hours to find the solution. Due to the short solution time, the model can quickly be updated and resolved periodically as more information about the future becomes available. Also, different instances of the same problem can be solved in a relatively short time to determine the effect of different reservoir simulations on the outcome.

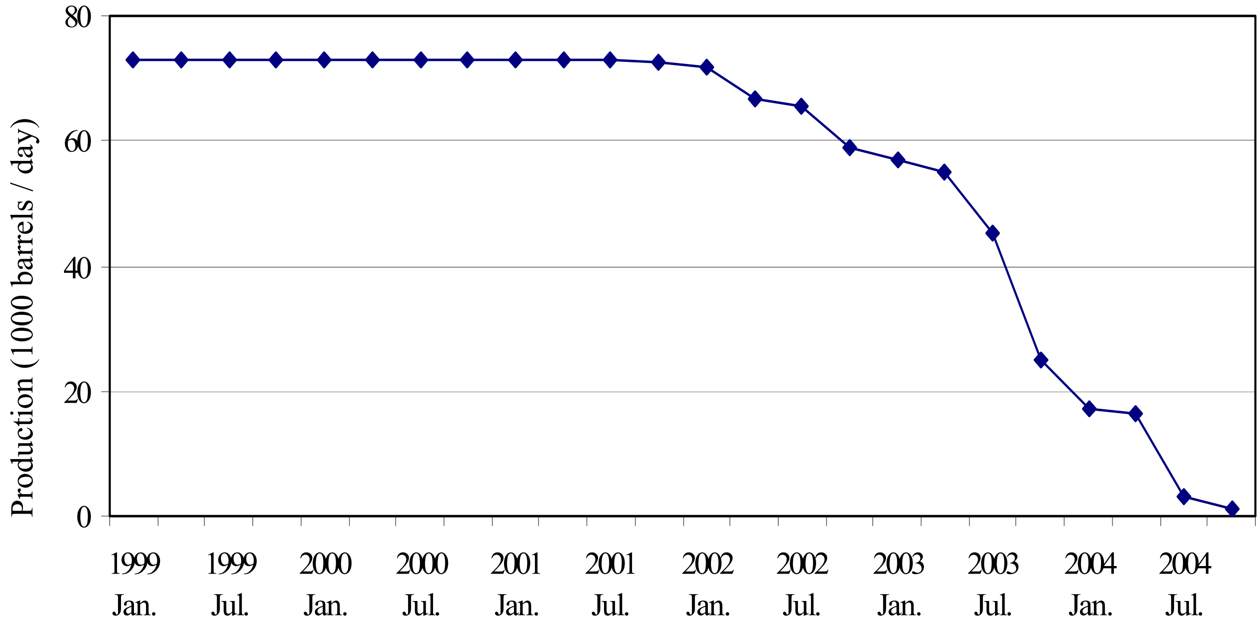


**Figure 10:** Production profile over six years.

## Conclusions

A multiperiod MINLP model for offshore oilfield infrastructure planning has been presented based on the linear model proposed by Iyer *et al*. (1998). By incorporating nonlinearities directly into the model, several thousands of binary interpolation variables were eliminated. Non-convexities resulting from the nonlinear equations were dealt with by using convex envelopes.

Furthermore, an iterative aggregation/disaggregation algorithm, in which logic-based methods, bilevel decomposition and aggregation of time periods are integrated, has been proposed for the solution of this model. A novel dynamic programming subproblem (PIA) has been proposed to update the aggregation scheme after each iteration in order to find a better bound from the aggregate problem.

This algorithm has been applied to an industrial sized example and results show significantly reduced solution times compared to a full space solution strategy, while the quality of the solution is similar to that of the full space solution. Results also show that model (PAI) leads to an aggregation scheme that resembles the disaggregate problem more closely, thereby giving a better indication of the quality of the solution.

More complex nonlinearities need to be incorporated in future work. Another possible extension of the application in future work is to deal with uncertainty in reservoir simulations and future economical factors.

## Acknowledgment

The authors would like to acknowledge financial support from Mobil Technology Company. Furthermore, the authors would like to thank Greger Ottosson for the useful hint and discussion on dynamic programming.

### References

Aranofsky, J.S., and Williams, A.C., "The Use of Linear Programming and Mathematical Models in Underground Oil Production," *Management Science*, Vol. 8, pp. 394-407, 1962.

Attra, H.D., Wise, W.B., and Black, W.M., "Application of Optimizing Techniques for Studying Field Production Operations," *don't know reference SPE.*

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

Bellman, R., *Dynamic Programming*, Princeton University Press.

Bohannon, J., "A Linear Programming Model for Optimum Development of Multi-Reservoir Pipeline Systems," *Journal of Petroleum Technology*, Vol. 22, pp. 1429-1436, 1970.

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

Dawson, R.G., and Fuller, J.D., "A Mixed Integer Nonlinear Program for Oilfield Production Planning," Working paper, 1996.

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

Eeg, O.S., and Herring, T., "Combining Linear Programming and Reservoir Simulation to Optimize Asset Value," paper SPE 37446 presented at the SPE Production Operations Symposium, Oklahoma City, 9-11 March, 1997.

Frair, L.C., "Economic Optimization of Offshore Oilfield Development," PhD Dissertation, University of Oklahoma, Tulsa, OK, 1973.

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.

Haugland, D., Hallefjord, Å., and Asheim, H., “Models for Petroleum Field Exploitation,” *European Journal of Operational Research*, Vol. 37, pp. 58-72, 1988.

Horst, R., and Pardalos, P.M., (eds.), *Handbook of Global Optimization*, Kluwer Academic Publishers, 1995.

Iyer, R.R., Grossmann, I.E., Vasantharajan, S., and Cullick, A.S., “Optimal Planning and Scheduling of Offshore Oil Field Infrastructure Investment and Operations,” *Ind. Eng. Chem. Res.,* Vol. 37, No. 4, 1998.

Lee, A.S., and Aranofsky, J.S., “A Linear Programming Model for Scheduling Crude Oil Production,” *Petroleum Transactions, AIME,* Vol. 213, pp. 389-392, 1958.

Nygreen, B., Christiansen, M., Haugen, K., Bjørkvoll, T., and Kristiansen, Ø, “Modeling Norwegian petroleum production and transportation,” *Annals of Operations Research*, Vol. 82, pp. 251-267, 1998.

Papalexandri, K.P., and Pistikopoulos, E.N., “A Multiperiod MINLP Model for the Synthesis of Flexible Heat and Mass Exchange Networks,” *Computers and Chemical Engineering*, 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 and Chemical Engineering*, Vol. 16, No. 3, pp. 189-210, 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.

Rogers, D.F., Plante, R.D., Wong, R.T., and Evans, J.R., “Aggregation and Disaggregation Techniques and Methodology in Optimization,” *Operations Research*, Vol. 39, No. 4, pp. 553-582, 1991.

Sullivan, J., “A Computer Model for Planning the Development of an Offshore Oil Field,” *Journal of Petroleum Technology*, Vol. 34, pp. 1555-1564, 1982.

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.

Van den Heever, S.A., and Grossmann, I.E., “Disjunctive Multiperiod Optimization Methods for Design and Planning of Chemical Process Systems,” Accepted for publication in *Computers and Chemical Engineering*, 1999.

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

Viswanathan, J.,and Grossmann, I.E., “A combned penalty function and outer-approximation method for MINLP optimization,” *Computers chem. Engng*, Vol. 14, pp. 769-782, 1990

Wilkinson. S.J., “Aggregate Formulations for Large-Scale Process Scheduling Problems,” Thesis Dissertation, Department of Chemical Engineering and Chemical Technology, Imperial College, London, 1996.

## Appendix A: Disjunctive model

The following is a complete mathematical model of the offshore oilfield infrastructure planning problem. The model is based on the one proposed by Iyer *et al*. (1998), with the main differences being that we include non-linearities directly into the model, do not consider the constraints for the movement of drilling rigs explicitly, and formulate the problem in disjunctive form. For ease of comparison, we use the same nomenclature and make the same assumptions as Iyer *et al*.. First, we present the sets, indices, variables and parameters. We omit superscripts in the variable definition for the sake of simplicity.

**Sets and Indices:**

| | |
|---|---|
| $PP$ | set of production platforms |
| $p$ | production platform $p \in PP$ |
| $WP(p)$ | set of well platforms associated with platform $p$ |
| $\pi$ | well platform $\pi \in WP(p)$ |
| $F$ | set of fields |
| $f$ | field $f \in F$ |
| $R(f)$ | set of reservoirs associated with field $f$ |
| $r$ | reservoir $r \in R(f)$ |
| $W_{WP}(\pi)$ | set of wells associated with well platform $\pi$ |
| $W_R(r)$ | set of wells associated with reservoir $r$ |
| $W_{WP,R}(r, \pi)$ | set of wells associated with reservoir r and well platform $\pi$ |
| $w$ | well $w \in W_{(.)}(.)$ |
| $t$ | time periods |
| $\tau$ | aggregated time periods |

To denote a specific well $w$ that is associated with a specific well platform $\pi$, which is in turn associated with a specific production platform $p$, we use the index combination ($w$, $\pi$, $p$). Similarly, the index ($\pi$, $p$) applies to a specific well platform $\pi$ associated with a specific production platform $p$.

**Continuous variables:**

$x_t$ oil flow rate in period $t$

$xc_t$ cumulative oil flow up to period $t$

$g_t$ gas flow rate (volumetric) in period $t$

$gc_t$ cumulative gas flow up to period $t$

$l_t$ oil flow (mass) in period $t$

$\phi_t$ gas-to-oil ratio (GOR) in period $t$

$v_t$ pressure in period $t$

$\delta_t$ pressure drop at choke in period $t$

$d_t$ design variable in period $t$

$e_t$ design expansion variable in period $t$

$Rev_t$ sales revenue in period $t$

$CI_t$ investment cost in period $t$ (including depreciation)

**Boolean variables:**

$z_t$ = true if facility (well, WP or PP) is drilled/installed in period $t$

**Parameters:**

$\rho$ productivity index of well

$P_{max}$ maximum pressure drop from well bore to well head

$GOR_{max}$ maximum $GOR$

$m_\tau$ number of periods in aggregate time period $\tau$

$T_a$ number of aggregate time periods, $\tau = 1 .. T_a$

$M_w$ maximum number of wells drilled in a time period

$\Delta t$ length of time period $t$

$U$ upper bound parameter (defined by the respective constraint)

$\alpha$ pressure drop coefficient for oil flow rate

| | |
|---|---|
| β | pressure drop coefficient for GOR |
| $c_{1t}$ | discounted revenue price coefficient for oil sales |
| $c_{2t}$ | discounted fixed cost coefficient for capital investment |
| $c_{3t}$ | discounted variables cost coefficient for capital investment |
| $\gamma_{p1}$ | first coefficient for pressure vs. cumulative oil |
| $\gamma_{p2}$ | second coefficient for pressure vs. cumulative oil |
| $\gamma_{g1}$ | first coefficient for cumulative gas vs. cumulative oil |
| $\gamma_{g2}$ | second coefficient for cumulative gas vs. cumulative oil |
| $\gamma_{g3}$ | third coefficient for cumulative gas vs. cumulative oil |
| $\gamma_{gor1}$ | first coefficient for GOR vs. cumulative oil |
| $\gamma_{gor2}$ | second coefficient for GOR vs. cumulative oil |
| $\gamma_{gor3}$ | third coefficient for GOR vs. cumulative oil |

**Superscripts:**

| | |
|---|---|
| $(w,\pi,p)$ | variables associated with well $w \in W$, with well platform $\pi$ and production platform $p$ |
| $(\pi,p)$ | variables associated with well platform $\pi$ and production platform $p$ |
| (p) | variables associated with production platform $p$ |
| $(r)$ | variables associated with reservoir $r$ |

**Objective function:**

The objective function is to maximize the Net Present Value (NPV) which includes sales revenues, investment costs and depreciation.

$$\max \Psi = \sum_{t=1}^{T} \{\mathrm{Re}\, v_t - \sum_{p \in PP} [CI_t^p + \sum_{\pi \in WP(p)} \{CI_t^{\pi,p} + \sum_{w \in W_{WP}(\pi)} CI_t^{w,\pi,p}\}]\} \qquad (1)$$

**Constraints valid for the whole infrastructure:**

In (2) the sales revenue in each time period is calculated from the total oil produced, which is in turn calculated in (3) as the sum of the oil produced from all production platforms. (4) calculates the amount of oil flow from each reservoir in each time period to be the sum of all oil flowrates from wells associated with that reservoir times the duration of the time period. The cumulative flow of oil from each reservoir is calculated in (5). Note that (5) is one of the linking constraints that links the time periods together and thus prevents a solution procedure where each period is solved individually. The cumulative flow of oil is used in (6) to calculate the reservoir pressure through the exponential function which is obtained by fitting a nonlinear curve to the linear interpolation data used by Iyer *et al*. (1998).

$$\mathrm{Re}\,v_t = c_{1t} x_t^{total} \tag{2}$$

$$\sum_{p \in PP} x_t^p = x_t^{total} \tag{3}$$

$$l_t^{r,f} = \Delta t \sum_{(w,\pi,p) \in W_{F,R}(f,r)} x_t^{w,\pi,p} \qquad \forall r \in R(f), f \in F \tag{4}$$

$$xc_\theta^{r,f} = \sum_{t=1}^{\theta-1} l_t^{r,f} \qquad \forall r \in R(f), f \in F \tag{5}$$

$$v_t^{r,f} = \gamma_{p1}^{r,f} \exp(\gamma_{p2}^{r,f} xc_t^{r,f}) \qquad \forall r \in R(f), f \in F \tag{6}$$

for t = 1..T

**Disjunction for each PP,WP and well:**

We exploit the hierarchical structure of the oilfield to formulate the disjunctive model. The production platforms are at the highest level of the hierarchy, and the disjunction includes all constraints valid for that PP, as well as the disjunction for the next hierarchical level, i.e. for all WPs associated with that PP. In turn, the disjunction for each WP, which is located within the disjunction of a PP, contains all constraints valid for that WP, as well as the disjunctions for all wells associated with that WP. We present the disjunctions here with numbers indicating the constraints present, and follow with the explanation of the individual constraints contained in each disjunction:

$$
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_{\theta}^{p} \\
\left[\begin{array}{c} z_{t}^{p} \\ (7),(8) \end{array}\right] \vee \left[\begin{array}{c} \neg z_{t}^{p} \\ CI_{t}^{p}, e_{t}^{p} = 0 \end{array}\right] \\
(9),(10),(11),(12) \\
\\
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p} \\
\left[\begin{array}{c} z_{t}^{\pi,p} \\ (13),(14) \end{array}\right] \vee \left[\begin{array}{c} \neg z_{t}^{\pi,p} \\ CI_{t}^{\pi,p}, e_{t}^{\pi,p} = 0 \end{array}\right] \\
(17),(18),(19) \\
\\
\left[\begin{array}{c}
\bigvee_{\theta=1}^{t} z_{\theta}^{w,\pi,p} \\
\left[\begin{array}{c} z_{t}^{w,\pi,p} \\ (20) \end{array}\right] \vee \left[\begin{array}{c} \neg z_{t}^{w,\pi,p} \\ CI_{t}^{w,\pi,p} = 0 \end{array}\right] \\
(21),(22),(23),(24),(25), \\
(26),(27),(28),(29)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_{\theta}^{w,\pi,p} \\
x_{t}^{w,\pi,p} = 0 \\
g_{t}^{w,\pi,p} = 0 \\
xc_{t}^{w,\pi,p} = 0 \\
gc_{t}^{w,\pi,p} = 0
\end{array}\right] \\
\forall w \in W_{WP}(\pi)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p} \\
x_{t}^{\pi,p} = 0 \\
g_{t}^{\pi,p} = 0
\end{array}\right] \\
\forall \pi \in WP(p)
\end{array}\right] \vee \left[\begin{array}{c}
\neg \bigvee_{\theta=1}^{t} z_{\theta}^{p} \\
x_{t}^{p} = 0 \\
g_{t}^{p} = 0
\end{array}\right]
$$

$$\forall p \in PP, t = 1..T$$

The outer disjunction is valid for each PP in each time period and can be interpreted as follows: If production platform $p$ has been installed during or before period $t$ (discrete expression $\bigvee_{\theta=1}^{t} z_{\theta}^{p}$ = true), then all constraints in the largest bracket are applied:

$$\begin{bmatrix} z_t^p = 1 \\ CI_t^p = c_{2t}^p + c_{3t}^p e_t^p \qquad (7) \\ e_t^p \le U \qquad (8) \end{bmatrix} \vee \begin{bmatrix} z_t^p = 0 \\ CI_t^p, e_t^p = 0 \end{bmatrix}$$

$$x_t^p \le d_t^p \qquad (9)$$

$$d_t^p = d_{t-1}^p + e_t^p \qquad (10)$$

$$\sum_{\pi \in WP_{(p)}} x_t^{\pi,p} = x_t^p \qquad (11)$$

$$\sum_{\pi \in WP_{(p)}} g_t^{\pi,p} = g_t^p \qquad (12)$$

First, the smaller nested disjunction is used to calculate the discounted investment cost (including depreciation) of the production platform in each time period. This cost is calculated if production platform $p$ is installed in period $t$ ($z_t^p = True$), otherwise it is set to zero. (7) relates the cost as a function of the expansion capacity, which is set to zero if the production platform is not installed ($z_t^p = False$), while (8) sets an upper bound on the expansion. (9) determines the design capacity to be the maximum flow among all time periods, and this is modeled linearly by defining the expansion variable which can take a non-zero value in only one time period. (11) and (12) are mass balances calculating the oil/gas flow from the PP as the sum of the flow from all WPs associated with that PP. If the production platform has not been installed yet (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^p$ = false), the oil/gas flows, as well as investment cost, are set to zero.

The middle disjunction is valid for all well platforms associated with production platform $p$ and is only applied if the discrete expression $\bigvee_{\theta=1}^{t} z_\theta^p$ is true. This disjunction states that if well platform π has been installed before or during period $t$ (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^{\pi,p}$ = true), then the constraints present in that disjunction are applied:

$$\begin{bmatrix} z_t^{\pi,p} \\ CI_t^{\pi,p} = c_{2t}^{\pi,p} + c_{3t}^{\pi,p} e_t^{\pi,p} \quad (13) \\ e_t^{\pi,p} \le U \quad (14) \end{bmatrix} \vee \begin{bmatrix} \neg z_t^{\pi,p} \\ CI_t^{\pi,p}, e_t^{\pi,p} = 0 \end{bmatrix}$$

$$x_t^{\pi,p} \le d_t^{\pi,p} \quad (15)$$

$$d_t^{\pi,p} = d_{t-1}^{\pi,p} + e_t^{\pi,p} \quad (16)$$

$$\sum_{\pi \in W_{WP}(\pi)} x_t^{w,\pi,p} = x_t^{\pi,p} \quad (17)$$

$$\sum_{\pi \in W_{WP}(\pi)} g_t^{w,\pi,p} = g_t^{\pi,p} \quad (18)$$

$$v_t^p = v_t^{\pi,p} - \alpha x_t^{\pi,p} - \beta g_t^{\pi,p} - \delta_t^{\pi,p} \quad (19)$$

Again, the smaller nested disjunction is used to calculate the discounted investment cost (including depreciation) of the well platform in each time period. This cost is calculated if well platform $\pi$ is installed in period $t$ ($z_t^{\pi,p} = True$), otherwise it is set to zero. (13) relates the cost as a function of the expansion capacity, which is set to zero if the well platform is not installed ($z_t^{\pi,p} = False$), while (14) sets and upper bound on the expansion. (15) and (16) determine the design capacity as described in the case of the production platform. (17) and (18) are mass balances calculating the oil/gas flow from the WP as the sum of the flow from all wells associated with that WP. (19) relates the pressure at the WP to the pressure at the PP it is associated with. The pressure at the PP is the pressure at the WP minus the pressure drop in the corresponding pipeline, which is given by the remaining terms in (19). If the production platform has not been installed yet (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^{\pi,p}$ = false), the oil/gas flows, as well as investment cost, are set to zero.

The innermost disjunction is valid for each well $w$ associated with well platform $\pi$, and is only included if well platform $\pi$ has already been installed (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^{\pi,p}$ = true). If well $w$ has been drilled during or before period t (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^{w,\pi,p}$ = true), then the following constraints are applied:

$$\begin{bmatrix} z_t^{w,\pi,p} \\ CI_t^{w,\pi,p} = c_{2t}^{w,\pi,p} \quad (20) \end{bmatrix} \vee \begin{bmatrix} \neg z_t^{w,\pi,p} \\ CI_t^{w,\pi,p} = 0 \end{bmatrix}$$

$$v_t^{\pi,p} = v_t^{w,\pi,p} - \alpha x_t^{w,\pi,p} - \beta g_t^{w,\pi,p} - \delta_t^{w,\pi,p} \tag{21}$$

$$x_t^{w,\pi,p} = \rho^{w,\pi,p}(v_t^{r,f} - v_t^{w,\pi,p}) \tag{22}$$

$$g_t^{w,\pi,p} \le x_t^{w,\pi,p} GOR_{\max} \tag{23}$$

$$x_t^{w,\pi,p} \le \rho^{w,\pi,p} P_{\max} \tag{24}$$

$$xc_t^{w,\pi,p} = \sum_{\theta=1}^{t-1} x_t^{w,\pi,p} \Delta t \tag{25}$$

$$gc_t^{w,\pi,p} = \sum_{\theta=1}^{t-1} x_t^{w,\pi,p} \Delta t \tag{26}$$

$$x_t^{w,\pi,p} \ge x_{t+1}^{w,\pi,p} \tag{27}$$

$$gc_t^{w,\pi.p} = \gamma_{g1}^{r,f} + \gamma_{g2}^{r,f} xc_t^{w,\pi.p} + \gamma_{g3}^{r,f} (xc_t^{w,\pi.p})^2 \qquad \forall (w,\pi,p) \in W_{F,R}(f,r) \tag{28}$$

$$GOR_t^{w,\pi.p} = \gamma_{gor1}^{r,f} + \gamma_{gor2}^{r,f} xc_t^{w,\pi.p} + \gamma_{gor3}^{r,f} (xc_t^{w,\pi.p})^2 \qquad \forall (w,\pi,p) \in W_{F,R}(f,r) \tag{29}$$

The smaller nested disjunction is used to calculate the discounted investment cost (including depreciation) of the well in each time period. This cost is calculated in (20) if well $w$ is drilled in period $t$ ($z_t^{w,\pi,p} = True$), otherwise it is set to zero. (21) relates the pressure at the well to the pressure at the WP it is associated with. The pressure at the WP is the pressure at the well minus the pressure drop in the corresponding pipeline, which is given by the remaining terms. (22) states that the oil flowrate equals the productivity index times the pressure differential between reservoir and well bore. (23) restricts the gas flowrate to be the oil flow times the GOR, while (24) restricts the maximum oil flow to equal the productivity index times the maximum allowable pressure drop. (25) and (26) calculate the cumulative flow to be the sum of flows over all periods up to the current one. Note that (25) and (26) are linking constraints that link the time periods together and prevent a solution procedure where every time period is solved separately. (27) denotes a specification by the oil company which restricts the flow profile to be non-increasing. The linear interpolation to calculate cumulative gas and GOR as functions of cumulative oil, are replaced by the nonlinear constraints (28) and (29). These quadratic equations are obtained from a curve fit of the linear interpolation data from Iyer *et al*. (1998). If the well has not been drilled yet (discrete expression $\bigvee_{\theta=1}^{t} z_\theta^{w,\pi,p}$ = false), the oil/gas flows, cumulative flows, as well as investment cost, are set to zero.

**Logical constraints:**

These represent logical relationships between the discrete decisions. (30) – (32) specify that each well, WP and PP can be drilled/installed in only one period. (33) states that if a WP has not been installed by *t,* then any well *w* associated with that WP cannot be drilled in *t*. Likewise, (34) states that if a PP has not been installed by period *t*, then any WP associated with that PP cannot be installed in *t*. The restriction that only $M_w$ wells can be drilled in any given time period, is given by (35).

$$\bigvee_{t=1}^{T} z_{\theta}^{w,\pi,p} \qquad \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP \tag{30}$$

$$\bigvee_{t=1}^{T} z_{\theta}^{\pi,p} \quad \forall \pi \in WP(p), p \in PP \tag{31}$$

$$\bigvee_{t=1}^{T} z_{\theta}^{p} \quad \forall p \in PP \tag{32}$$

$$\neg \bigvee_{\theta=1}^{t} z_{\theta}^{\pi,p} \Rightarrow \neg z_{t}^{w,\pi,p} \qquad \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP \tag{33}$$

$$\neg \bigvee_{\theta=1}^{t} z_{\theta}^{p} \Rightarrow \neg z_{t}^{\pi,p} \qquad \forall \pi \in WP(p), p \in PP \tag{34}$$

$$\underset{(w,\pi,p)}{\vee} z_{t}^{w,\pi,p} \leq M_{w} \tag{35}$$

for t = 1..T.

## Appendix B: Derivation of the convex hull for problem (MP)

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 (MP) to its mixed integer form through the convex hull formulation of each disjunction. The convex hull gives a tighter formulation than the "big-M" formulation (Turkay and Grossmann, 1996b). 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. In the formulation given below, superscript 1 refers to the left disjunction, while superscript 2 refers to the right disjunction for each set of disjunctions. 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. The disaggregate variables that go to zero become redundant and are removed to simplify the model. For simplicity, a binary variable $y_t$ is assigned to each disjunction in place of the Boolean expression $\bigvee_{\theta=1}^{t} z_\theta$. After applying the convex hull to each disjunction of (MP) the MINLP formulation (MIP) is as follows:

$$\max \Psi(x,z)$$

subject to

$$
\begin{array}{ll}
Ax_t \le b & \forall t = 1..T \\
f_t^{lin,k}(x_t) \le 0 & \forall t = 1..T \\
f_t^{env}(x_t) \le 0 & \forall t = 1..T \\
g_t(x_1, x_2 .. x_t) \le 0 & \forall t = 1..T \\
x_{t,nz} = x_{t,nz}^1 + x_{t,nz}^2 & \forall t = 1..T \\
B_p x_{t,z}^p + C_p x_{t,nz}^{p,1} \le b_p y_t^p & \forall p \in PP, t = 1..T \\
x_{t,z}^p \le U y_t^p & \forall p \in PP, t = 1..T \\
B_\pi x_{t,z}^{\pi,p} + C_\pi x_{t,nz}^{\pi,p,1} \le b_\pi y_t^{\pi,p} & \forall \pi \in WP(p), p \in PP, t = 1..T \\
x_{t,z}^{\pi,p} \le U y_t^{\pi,p} & \forall \pi \in WP(p), p \in PP, t = 1..T \\
A_w x_{t,z}^{w,\pi,p} + A_w x_{t,nz}^{w,\pi,p,1} \le b_w y_t^{w,\pi,p} & \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP, t = 1..T \\
g_{t,w}(x_{1,z}^{w,\pi,p}, x_{2,z}^{w,\pi,p} .. x_{t,z}^{w,\pi,p}) \le U y_t^{w,\pi,p} & \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP, t = 1..T \\
h_{t,w}^{lin,k}(x_t^{w,\pi,p}) = \nabla_{x_z^j} g(x^{j,k})(x_z^j) + \nabla_{x_{nz}^j} g(x^{j,k})(x_{nz}^{j,1}) - x_z^i - x_{nz}^{i,1} & \\
\quad \le \left[ - g(x^{j,k}) + \nabla_{x_z^j} g(x^{j,k})(x_z^{j,k}) + \nabla_{x_{nz}^j} g(x^{j,k})(x_{nz}^{j,k}) \right] y_t^{w,\pi,p} & \\
 & \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP, t = 1..T \\
h_{t,w}^{env}(x_t^{w,\pi,p}) = x_z^i + x_{nz}^{i,1} + \left( \dfrac{g(x_U^j) - g(x_L^j)}{x_U^j - x_L^j} \right) \left( x_z^j \right) + \left( \dfrac{g(x_U^j) - g(x_L^j)}{x_U^j - x_L^j} \right) \left( x_{nz}^{j,1} \right) & \\
\quad \le \left[ g(x_L^j) + \left( \dfrac{g(x_U^j) - g(x_L^j)}{x_U^j - x_L^j} \right) \left( x_L^j \right) + \left( \dfrac{g(x_U^j) - g(x_L^j)}{x_U^j - x_L^j} \right) \left( x_L^j \right) \right] y_t^{w,\pi,p} & \\
 & \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP, t = 1..T \\
x_{t,z}^{w,\pi,p} \le U y_t^{w,\pi,p} & \forall w \in W_{WP}(\pi), \pi \in WP(p), p \in PP, t = 1..T \\
h(y) \le 0 & \\
x \ge 0, \quad y \in \{0,1\} &
\end{array}
$$

## Appendix C: Illustrative example for the application of model (PAI)

Sub-problem (PAI) is used to determine the aggregation scheme which yields an upper bound that is as close as possible to the lower bound from the disaggregate solution. The input for (PAI) is therefore the solution from the previous disaggregate sub-problem. Consider such a solution for a disaggregate sub-problem that was solved for 3 years, each divided into 4 quarterly periods, i.e. a total of $T$ = 12 time periods. We want to aggregate this problem into $TA$ = 3 periods. The data from the disaggregate solution that is used as input for (PAI) is as follows:

**Table 6:** Data for illustrative example of model (PAI)

| T | $f_{inv}$ | $f_{rev}$ | $f_{depr}$ | I ($mil.) | R ($mil.) |
|---|---|---|---|---|---|
| 1 | 1 | 0.981 | 0.529 | 55 | 5.3 |
| 2 | 0.981 | 0.962 | 0.49 | 0 | 5.3 |
| 3 | 0.962 | 0.943 | 0.45 | 0 | 5.3 |
| 4 | 0.943 | 0.925 | 0.408 | 0 | 5 |
| 5 | 0.925 | 0.907 | 0.367 | 0 | 5 |
| 6 | 0.907 | 0.889 | 0.324 | 10 | 4.8 |
| 7 | 0.889 | 0.872 | 0.28 | 0 | 4.5 |
| 8 | 0.872 | 0.855 | 0.236 | 0 | 4 |
| 9 | 0.855 | 0.838 | 0.19 | 0 | 1 |
| 10 | 0.838 | 0.822 | 0.144 | 0 | 1 |
| 11 | 0.822 | 0.806 | 0.097 | 0 | 0.5 |
| 12 | 0.806 | 0.79 | 0.049 | 0 | 0.1 |

For the above disaggregate solution, an objective value of $ 6.173 million is obtained, which is a lower bound to the actual optimal solution. To determine the aggregation scheme that would provide an upper bound that is as close as possible to this lower bound, (PAI) is solved and an objective value of $ 7.798 million is obtained for an aggregation scheme of 1,4,7. The next step in the algorithm is to solve an aggregate subproblem which will now be aggregated with this new scheme of 1,4,7 and the user knows to expect an upper bound in the order of $7.798 million. Note that if the user had blindly aggregated with a scheme of 4,4,4, which is intuitive since this would represent aggregating 12 quarters into 3 years, an upper bound of $ 9.332 million would be obtained. This represents a gap of 51.2% between bounds, versus the 26.3 % gap obtained by the improved aggregation scheme obtained from (PAI).