Note on simulator-based optimization for petroleum production optimization

Stine Ursin-Holm, Vidar Gunnerud, Bjarne Foss and Anders Sandnes

Abstract

In this paper, we present an optimization approach that explores the natural structure of a network simulator, when searching for the best possible settings given a predefined objective and constraints. The approach is tested on a petroleum production network, where it significantly outperforms the standard industry approach, both regarding solution time and solution quality.

The approach, which the authors has named SmartOpt, splits the optimization variable space into a discrete and a continues sub-space, and the network simulator into "simulator components", one for each component in the network. The main benefits of the approach are twofold. First, the optimization variables can be treated by a branch and bound algorithm in combination with a gradient based search algorithm, while as a standard approach is forced to use a derivative free algorithm. Second, the simulator components are of such low dimension that it is possible to sample the search space up front and build fast proxy models that can be used in the optimization search.

As the approach opens up for many alternative strategies, the authors derive and analyze six alternatives, and compare them with a standard industry approach.

1. Introduction (to daily petroleum production optimization)

Automation is a keyword across a wide field of applications. Technology has developed rapidly in parallel, and computer power and the amount of data available from sensors is increasing every year. This has opened an opportunity window for automating processes that today might require significant human intervention. Within many applications, there is still a long way to go before full automation is possible, if ever. However, the level of automation can be increased by the development of advanced decision support with the use of complex mathematical models to guide the humans through these decision processes.

In this chapter, we suggest a division of the decision process into a sequence of steps. Based on this sequence we indicate the current level of automation within industry, monitoring and improvement workflows. Furthermore, we discuss the role of complex simulators and optimization algorithms within decision processes, and the influence on automation levels due to such tools. Lastly, we introduce decision-making within daily petroleum production, and the crying need for advanced decision support tools within this domain.

From sensor data to decisions

High quality decision making in daily operations is key to secure safe and efficient operations of complex production systems. Decision support systems are an important ingredient to secure high quality decisions, and the use of mathematical models and optimization is slowly penetrating into such systems. Decision support systems for operational decisions can be divided into a sequence of components as illustrated in Figure 1. Essentially, it starts by acquiring the relevant real-time information and ends in making and implementing a decision.

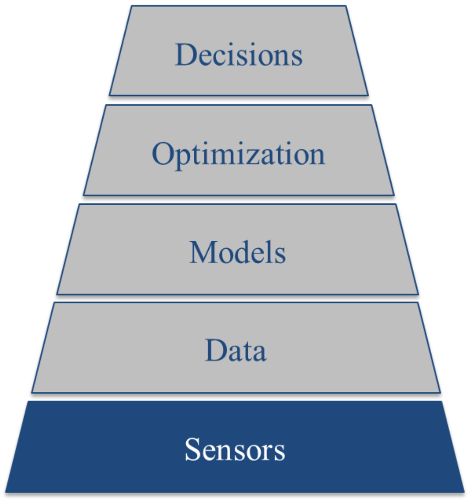

Figure 1 – Decision pyramid

To elaborate on the decision pyramid in Figure 1, at the bottom sensors capture real-time information about the system. This information is typically collected and stored in real-time databases. Further, when considering and evaluating alternative decisions, it is necessary to have an idea of how the system will respond to changes, thus data combined with (physical) understanding form the basis for building models with which one can produce simulation outputs. These may again be applied in what-if studies or, more ambitiously, for optimization where an optimizer will provide recommendations for the users on how to select control input values.

If similar decisions are made frequently, one may consider automating parts or all of this sequence. Automating the whole sequence would result in an "autopilot", i.e., an automated decision loop. This approach is, however, not feasible in many cases due to insufficient data quality, imprecise model or the need for humans to scrutinize, and possibly change, a recommendation before it is implemented. The level of automation one should aim for within the decision pyramid is thus problem specific.

If we limit the scope to the process and oil and gas industries, monitoring and improvement workflows linked to real-time data analyses involve a significant amount of manual work. However, computer models, usually in the form of simulators, help engineers analyze and reason about system behavior, and make it possible to predict the effect of various decisions. In many cases, models are essential because it is impossible for humans to sensibly communicate and reason about complex systems without some suitable level of abstraction. Around us, we see more and more simulators that are essentially "automating" the evaluations necessary for the prediction of system behavior.

If we move further up the decision pyramid, optimization systems compute recommendations based on an articulated objective function, constraints and the simulator model. The complexity increases exponentially, as two complex computer algorithms are combined, simulators and optimization algorithms, to compute an optimal decision. Today, these methods often fail to provide satisfactory results when the systems become too complex, and thus we do not find too many successful applications yet.

The runtime will affect the level of integration of such algorithms within a decision process. If the runtime is long (hours), the decision support may still provide highly valuable information, e.g., once a day. However, if the solutions are rapidly available (minutes), a decision support system may be used iteratively during, for example, decision meetings or in between other scheduled events.

Adding efficient real-time decision support to the human expertise and further automate the fourth layer to provide automatic generation of recommendation enables improved decision quality since it increases the robustness and reliability of the decision making process. It enables the engineers to focus on the top layer in the decision pyramid, and thus spend more time on more high-level considerations rather than lower level analyses.

In this paper we outline a concept, SmartOpt, which we believe can contribute with significant improvements in the decision-making process, enabling efficient optimization on top of network simulators. Previous publications on the concept are (Gunnerud & Foss, 2010), (Gunnerud, et al., 2013) and (Ursin-Holm, et al., 2014).

Simulation based optimization

Simulators have entered many engineering disciplines, no doubt because of their contribution in modelling complex systems. However, detailed simulators are often complex in terms of size and the phenomena modelled, and thus they may require excessive runtime to compute a solution with the required accuracy. Complexity typically relates to the number of model equations and/or the number of unknowns. This number may be high due to a large number of model components, as for instance in a simulator of a complete refinery, or because the simulator is based on discretization of partial differential equations using finite elements or some other appropriate method. The number of modelled phenomena and their inherent interaction also contributes to the complexity.

The industry uses simulators in a variety of ways; the most interesting in terms of this thesis is in "what-if" analyses of different solutions or alternative courses of action. If derivatives are available, they can be readily used for sensitivity analyses. Moreover, the models can usually scale up reasonably well, and so optimization can be applied to large problems. The importance of accurate derivatives for most optimization algorithms cannot be emphasized enough, as they essentially point the search in the right direction (Gunnerud et al., 2013), the text in the rest of this section is copied from this paper).

One particular area of interest is combining state-of-the art optimization techniques with the simulation of large-scale oil fields, possibly with the aim to optimize the production strategy from a producing oil field using updated reservoir models (see for example (Jansen, et al., 2008) and (Foss, 2012) for overviews). Typically, this consists of history matching along with some incorporation of additional data to, at least partially, resolve the underdetermined nature of the model and thereby adjust the reservoir model parameters. This is done in an attempt to reproduce the historical behavior, such as production rates and pressures, of the real reservoir, along with production optimization to optimize the future production strategy (see for example (Naevdal, et al., 2006)). Furthermore, there is an increasing interest in applying simulators for online studies. In the case of dynamic simulators, this may include predictive simulations where the simulator is used as a decision support tool in operation of, for instance, a supply chain, a power plant or a process plant. Online use of simulators requires online estimation functionality to reconcile the simulator with the available online data in real-time.

Detailed simulators are in daily use in a variety of applications. Hence, considerable capital and human resources have been invested in this technology. Furthermore, high-fidelity simulators require significant runtime to compute outputs. The SmartOpt concept developed by the authors embeds optimization into existing simulator software packages. The motivation for this is the fact that "what-if" analyses alone may be both inadequate and inefficient in decision situations, for example in optimizing continuous variables like well choke openings. A simulator combined with an optimization algorithm with recommendation capabilities, however, may improve decision quality as well as decision speed. This is becoming an ever more pervasive need in many different applications. From a practical point of view, one should not underestimate the importance of uniform viable databases for the input, and user-friendly appropriate output. Optimization technology and its use have developed immensely during the last couple of decades due to algorithmic advances as well increases in computing power. Huge problems with millions of continuous variables and thousands of discrete variables can be solved. (Dür, 2001) (Tuy, 2005). However, optimization problems with embedded simulators are particularly challenging. In such cases, the simulator is a function whose explicit form is often unknown, which computes some output measures based on input parameters. The number of function evaluations must be very limited if it is time-consuming to compute one solution of the simulator. Furthermore, simulators usually come as a "black box calculators" without the ability to compute gradients, which are usually crucial for fast convergence of an optimization algorithm to a (local) solution. It is for these reasons that simulation and optimization have been combined infrequently, at least until the present decade (Fu, 2002). This situation is however changing rapidly and there is a steady increase in papers on simulation combined with optimization.

The scientific literature on simulation-based optimization contains numerous applications. A few examples include supply chain management (Schwartz, et al., 2006) and (Wan, et al., 2005), combustion engine design (Jakobsson, et al., 2009), process system design (Jaluria, 2009) and oil field operations (Echeverría Ciaurri, et al., u.d.) and (Echeverría Ciaurri, et al., 2011). As may be expected the maturity of the applications vary a lot.

The SmartOpt concept is in this paper described in the context of decision making within field operations and petroleum production, more specifically for efficient daily petroleum production optimization as a valuable add-on to existing simulator software packages.

Daily petroleum production optimization

Decisions on the day-to-day production strategy are one of the key challenges in petroleum field operations. Oil companies themselves manage the daily oil production and make strategies for improvement. Several tools and individuals, both on- and offshore, are involved in the search for the best, or appropriate, production strategy, and decisions are often taken in interdisciplinary meetings between the production engineers and offshore operators. Typically, the production engineers wish to optimize the daily production of hydrocarbons from each well and the field as a whole, given topside restrictions, e.g., separators, compressors capacities etc., while the long-term drainage strategy of the reservoir are set by the reservoir engineers. Operators on the other hand want to minimize the amount of changes done to the system to keep things stable.

Updates of the production strategy usually occur at a pace that reflects the field dynamics. This typically means that a short-term production strategy has a limited lifetime, typically hours up to a few days, during normal operating conditions. Based on the current field conditions the engineers and operators arrive at a new production strategy for how to adjust the inlet pressures and chokes, operate gas lift valves if necessary, choose the routing of wells through pipelines and so on. Extraordinary events, such as equipment failure or replacements, will also trigger the need to find a new production strategy.

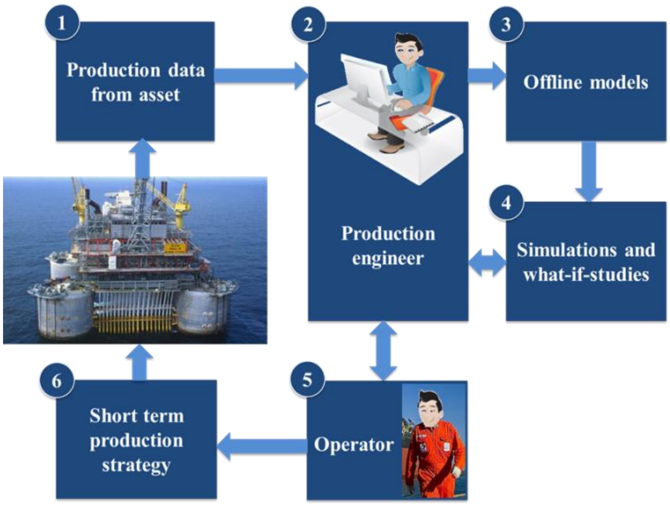

Figure 2 - The work process loop today

Figure 2 illustrates the typical key components involved in a daily work process loop. Initially the production engineers study the production history during the last week to identify changes in well performance. Models of the system are then updated according to the state of the field before they are used to guide the search for an improved production strategy. Production engineers typically conduct the daily "production optimization" through experience and decision support tools such as network simulators and what-if studies, to derive at a new production strategy. Discussions and meetings with the offshore operators are then conducted before deciding upon the final strategy, which the offshore operators will implement.

State-of-the-art decision support tools are frequently imprecise and slow at computing recommendations. SmartOpt however, represent efficient production optimization as an add-on to existing network simulator software, providing quick and reliable suggestions for optimal production strategies. It is applicable both before, and iteratively, during meetings between engineers and operators, which facilitates derivation of optimal strategies on which they both agree. In order to explain the SmartOpt concept and to discuss its benefits and possible challenges, a floating production, storage and offloading (FPSO) unit will be used as an illustrative example throughout the paper.

2. The production system

This chapter presents terminology to provide a deeper understanding of properties, production principles and dynamics of an oil field. We begin with an explanation of reservoir and well terminology, then further details about the production principles and issues are discussed.

System description

A reservoir is a rock body where hydrocarbons of several types such as natural gas, condensates, liquid hydrocarbons and water are stored (Alpha Thames Ltd, 2004). The three basic types of reserve phases are gas, oil and water, typically occupying different parts of the reservoir. Production systems are developed with the purpose of extracting the majority of the oil and gas reserves. Extraction of fluids causes changes in the composition of oil, gas and water within the reservoir. Gas and water production are often described through volume fractions. The ratio of produced gas to produced oil is called the Gas to Oil ratio (GOR) whereas the fraction of water in the liquid is referred to as Water Cut (WC).

Figure 3 is an illustration of a typical offshore production system. Hydrocarbon will flow from the *reservoir* into the *wellbore*, i.e. the bottom part of the well. The *tubing* directs the flow through the well from the wellbore to the *wellhead*. At the seabed, the s*ubsea wells* are connected to a *subsea manifold* that connects several subsea wells to through the same *riser pipelines*. For each well at the subsea manifold, a decision has to be made about where the flow is to be routed, since there often is more than one pipeline leaving each manifold. *Satellite wells* capture the hydrocarbons in the outskirts of a reservoir and are directly connected to the platform. In this paper the terms *satellite well* and *topside well* are used interchangeably. At the platform, the satellite wells and the riser pipes are routed to *separators* for processing.

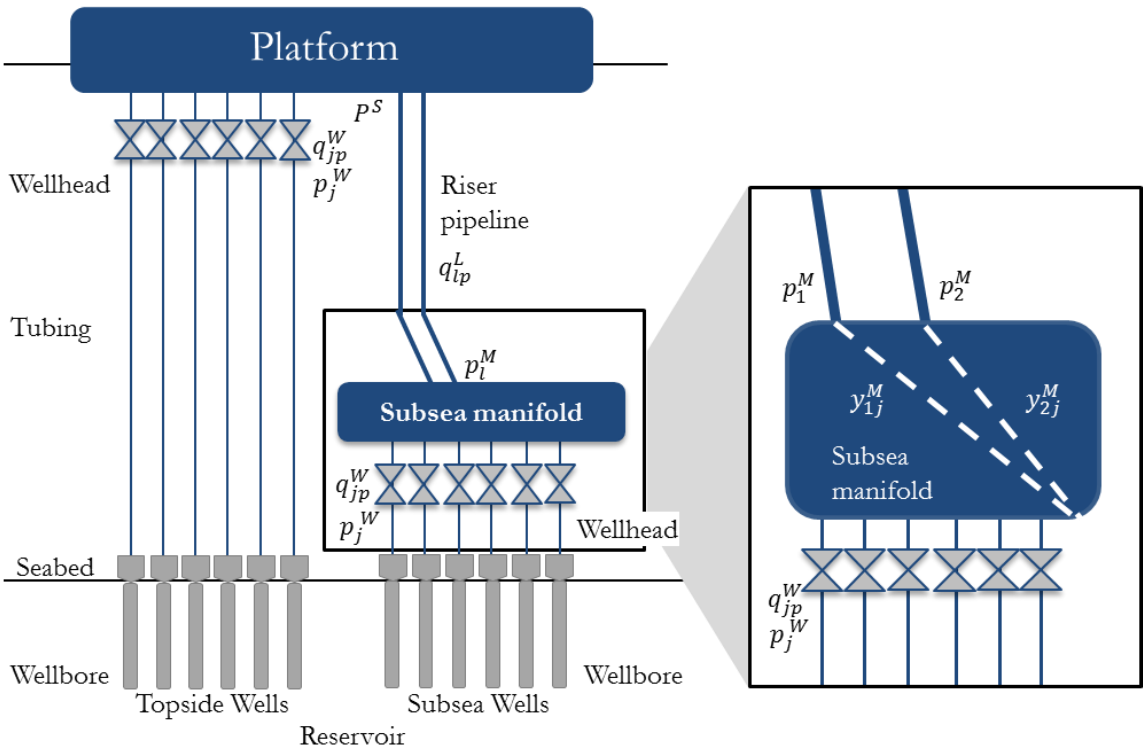

Figure 3 - Production system topology

Normally reservoir dynamics are slow with minimal changes over weeks and months, so the properties and compositions within the reservoir change gradually over the production lifetime. It may take months before significant changes in the GOR and WC occur, typically resulting in an increasing share of gas and/or water over time. If the reservoir dynamics are in-fact slow, GOR and WC can be assumed constant for the short-term planning period with a time horizon of days up to a week. In addition pipeline dynamics are fast and the flow of oil, gas and water through pipelines can be assumed to be steady-state. As a result, when dealing with a short-term production period the influence of time can be neglected.

Production principles

Hydrocarbons will flow from the reservoir to the topside as long as the pressure differentials are large enough. Starting from the boundary of the drainage area, formation fluids first flow through the porous media surrounding the well, and enter the wellbore. From here, vertical or inclined flow occurs in the well until the well stream reaches the seabed. Finally, the flow line transports the fluid to the separator holding a constant pressure at the topside process facility. Pressure losses occur through the system for several reasons.

Pushing the flow from the reservoir into the wellbore requires a lot of work and induces a pressure drop. The well flow rate depends on differential in the *reservoir pressure* and the *bottomhole pressure* experienced at the wellbore. Large pressure drop occurs when the flow is elevated through the system due to decline in the hydrostatic pressure and friction between the pipe wall and the production flow. Thus, for producing wells the bottomhole pressure must be larger than the *wellhead pressure*. To support liquid flow towards the platform, the wellhead pressures cannot fall below the *manifold pressure* or the *separator pressure,* for subsea wells and satellite wells respectively. Furthermore, the manifold pressure must be sufficiently high to drive the flow all the way through the riser pipes to the separator. The production engineers are able to control the day-to-day flow rates by using the *choke valves* or chokes. The choke valve regulates the rate of flow and reduces the wellhead pressure.

Artificial gas lift technology can be installed to lift more fluids from the reservoir when the wellhead pressure is not sufficiently high to support liquid flow up towards the platform. Gas is injected into the tubing to reduce fluid density, resulting in increased pressure drop between the reservoir and the wellbore and thus higher flow rates. Gas lift might be a restricted resource and should only be allocated to the wells that have fully open chokes and insufficient natural lift. Whenever a well is choked back, the natural lift is adequate and gas lift not really needed.

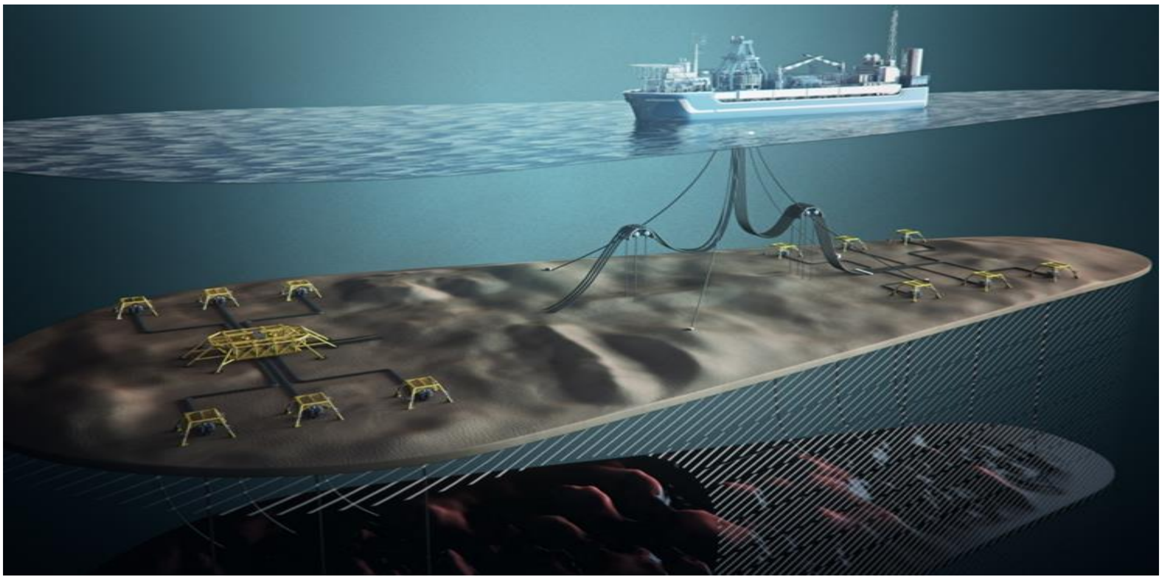

Figure 4 – Conceptual layout of the production system

Case study

The FPSO have six subsea and six platform wells. The model representing the wells are based on data from a Petrobras FPSO, and the pipeline model is based on data from a Statoil platform. All wells have the option to be gas lifted. The case is realistic and is considered common both in size and complexity. Figure 4 illustrates the production system setup.

It is assumed that production is off plateau and that the objective is to maximize the total oil production from day to day, by finding the optimal choke settings and gas lift rates for each well. The subsea wells must be routed to one of the two production lines or alternatively shut off. Topside capacity for water and gas handling and gas available for gas lift, is limited. Within the timeframe of this short-term optimization, we model the wells with constant GOR and WC.

The following explanation emphasizes the complexity of the problem at hand. Several subsea wells are usually producing to a pipeline simultaneously. Changes done to one of the wells routed to the production line, will affect the production of the other wells routed to the same line, even if the settings of these wells remain unchanged. E.g. some wells are produced with artificial gas lift. More artificial gas lift applied to a well will increase production from this well, but result in higher backpressure and lower production from the other wells. The total effect of a change in gas lift rate to one well is thus quite challenging to figure out since it is necessary to consider its effect on all the wells routed to the same production line. Wells producing with artificial lift usually have fully open chokes. However, if a well is producing naturally the choke valve might not be fully open. In this case, the change in manifold line pressure can be compensated for by adjusting the well's choke setting and thereby keeping the wellhead pressure and well production the same.

3. Production system modeling

Dynamic simulators are frequently used within the upstream oil and gas industry to model process systems and analyse pressures and flow behaviours throughout the processes. A dynamic simulator solves the mass and energy balances in a process system to obtain a rigorous description and the systems time-varying behaviour. This can be done by decomposing the process into rigorously modelled unit operations while robust numerical methods, equation solving and implicit integration solves the resulting process system.

When modelling a petroleum production system two aspects are particularly complicated. These are the reservoir fluid behavior in the link between the reservoir and the wellhead and the multiphase flow in pipelines. The importance of proper description of flow behavior in the wells and pipelines cannot be overemphasized because the pressure drop in well and pipes are the greatest and most decisive parts of the total pressure drop of the system. The reason for this is the great elevation difference and the consequently high hydrostatic term in the pressure gradient equation.

Inflow

Within process simulators, reservoirs are usually represented as a boundary condition with a fixed reservoir pressure and fluid composition. The Inflow Performance Relationship (IPR) describes inflow from the reservoir to the wellbore, in terms of the reservoir pressure and the well bottomhole pressure. The simplest approach to describe the inflow performance of oil wells is the use of the productivity index (PI) concept. This concept is based on Darcy's law, and is represented by equation (1) which states that the liquid inflow into a well is directly proportional to pressure drawdown. The concept was developed under the assumption that only a single-phase liquid is present in the reservoir near the wellbore. If the reservoir pressure, P^{RES}, is known, the wells PI can be found obtaining oil flow rates, q_{OIL}, for different bottom-hole pressures, p^{BHP}.

$$q_{OIL} = PI(P^{RES} - p^{BHP}) \tag{1}$$

If the reservoir pressure is below bubble point gas will also be present near the wellbore, and the assumptions used to develop PI is no longer valid. In this situation, a larger-than-linear pressure drop is required to increase production rates. This shape is approximated by Vogel's equation (2). If the reservoir pressure, P^{RES}, a single stabilized oil flow rate, q_{OIL}, and the corresponding bottomhole pressure, p^{BHP}, is known it is possible to construct the wells IPR curve (Takacs, 2005) (Gunnerud & Langvik, 2007).

$$\frac{q_{OIL}}{Q_{OIL,max}} = 1 - 0.2\left(\frac{p^{BHP}}{p^{RES}}\right) - 0.8\left(\frac{p^{BHP}}{p^{RES}}\right)^2 \quad (2)$$

Multiphase flow through pipelines

Multiphase flow in pipelines is characterized by different flow patterns according to velocity, composition and gas slip effects. Gas slip is the fact that the different phases move with different velocities which causes the mixture density to increase in comparison to the no-slip case. A selection of flow types are shown in Figure 5. Stratified flow occurs in horizontal pipes with low flow rates of liquid and gas. Pipes and wells can occur at any inclination. The stratified to intermittent transition is very sensitive to the pipes angle. At low to medium gas flow velocities in vertical pipes, the gas phase takes the form of uniformly distributed discrete bubbles rising in the continuous liquid phase. The gas tend to overtake the liquid particles and gas slippage occurs. If liquid velocities are high in relation to the gas velocities the gas bubbles exist in smaller bubbles evenly distributed in the continuous liquid phase moving at high velocity, this is called dispersed bubble flow. The phases now travel with similar velocity and no slippage effect occurs. Increased gas rates may lead to slug flow where the liquid slugs and large gas bubbles follow each other in succession. With higher gas rates the flow gets erratic, this phenomenon is called churn flow. At extremely high gas flow velocities annular flow occurs, where the gas flows in the middle of the pipe and pushes the fluid outwards against the pipe wall. Some of the liquids will be ripped loose from the walls and can be seen as small bubbles of liquid in the gas phase.

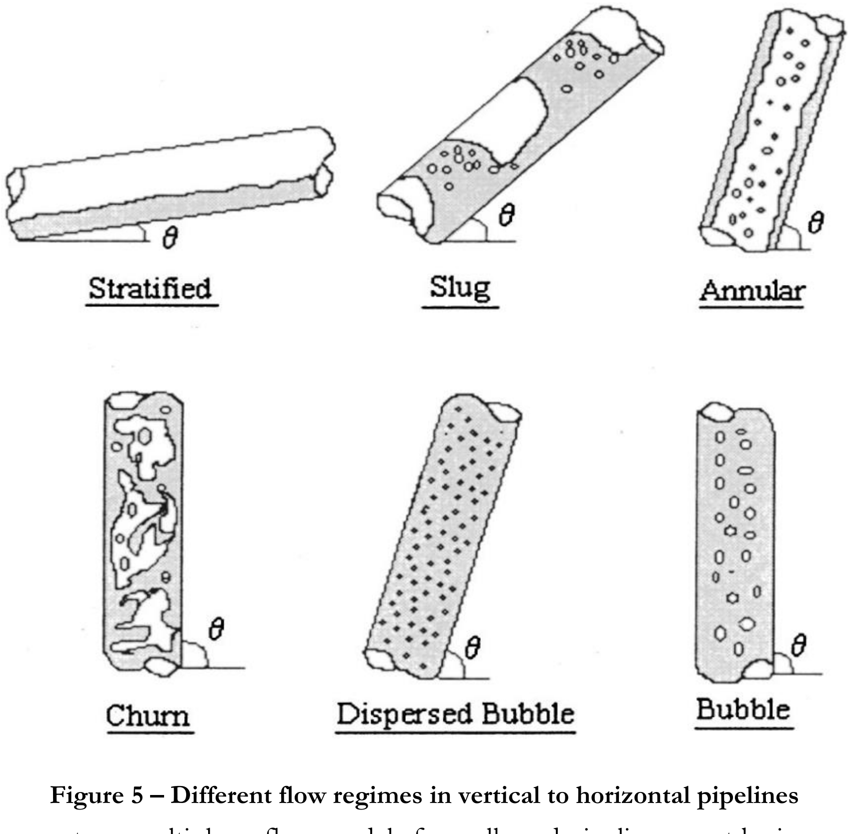

Figure 5 – Different flow regimes in vertical to horizontal pipelines

When a production system, multiphase flow models for wells and pipelines must be incorporated into the complete system model. Multiphase flow through pipelines is complex to model, phenomena and pressure drop varies greatly with flow regimes. The pressure reductions experienced when hydrocarbons flow through the wells and pipelines, equation (3), are partly due to friction and partly due to increased elevation. In addition, due to gas expansion as the pressure reduces pressure losses due to acceleration are induced (Takacs, 2005) (Gunnerud & Langvik, 2007).

$$\left(\frac{dP}{dL}\right)_{tot} = \left(\frac{dP}{dL}\right)_{friction} + \left(\frac{dP}{dL}\right)_{gravity} + \left(\frac{dP}{dL}\right)_{acceleration} \tag{3}$$

Well performance

The flow in the vertical part of the well is described by Vertical Lift Performance Curves (VLP). The VLP curves are correlations representing the pressure drop in the vertical part of the well, from the wellbore to the manifold. These curves relate the flow rate in the pipe to the wellhead pressure and the bottomhole pressure of the well.

IPR describes the inflow at a certain bottomhole pressure, and the VLP gives the correlation about pressure drop in the vertical part of the well, this relation gives a unique wellhead pressure at a certain inflow. Thus, the intersections IPR and the VLP curves present the production point with rates and bottomhole pressures for a certain wellhead pressures. This is show in figure Figure 6 (Gunnerud & Langvik, 2007).

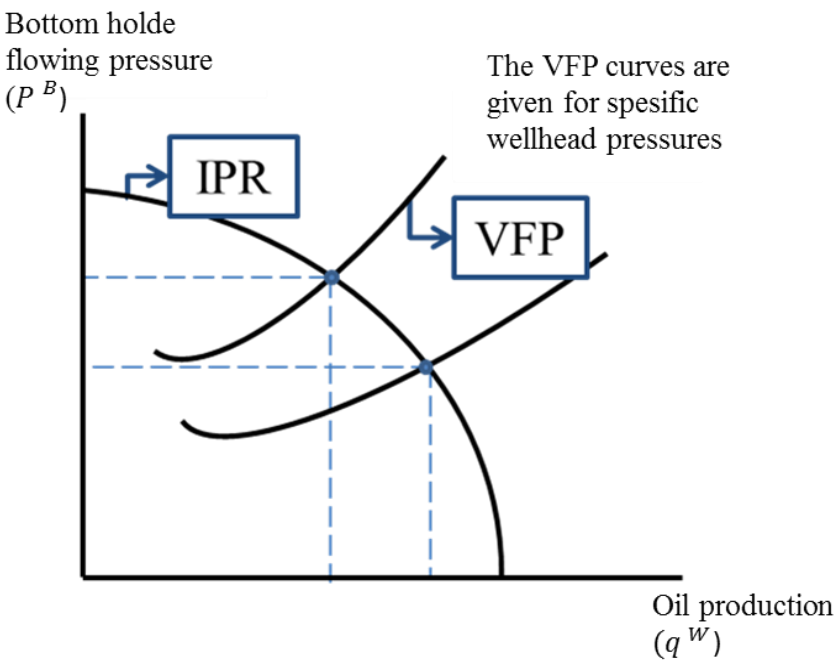

Figure 6 – Illustration of an Inflow Performance Relationship curve for a certain reservoir pressure intersecting two Vertical Flow Performance curves for different wellhead pressures respectively

4. Mathematical optimization

In this chapter, we present several strategies for daily petroleum production optimization as a valuable add-on to existing simulator software packages. We will start to outline the optimization problem formulation by describing what we call the "standard approach" to attack this problem. This is a non-invasive approach where the production network simulator is treated as one black box, only allowing the optimization algorithm to interrogate the simulator by requesting outputs based on a set of inputs, illustrated in Figure 5. The main challenge of this approach accrues when a (small) part of the variables space is discrete and the simulator only accepts discrete values for these variables. In this case, the complete variable space must be solved with a derivative free optimization algorithm considerably slowing down the convergence, compared to a gradient-based search.

Figure 7 – Industry standard optimization loop

Motivated by this challenge, we introduce the SmartOpt concept. We start to describe the concept by splitting up the decision variable space into a discrete part, which is solved by a derivative free method, and a continuous part, which is solved by a gradient search approach. Further, we describe how it is possible to decompose the network simulator into standalone simulators for each well and pipeline, such that mass and energy balances become explicit algebraic constraints within the optimization problem. As a result of this approach, the discrete decisions are no longer contained in the simulator codes, solely within the algebraic equations. This again enables us to relax the integer requirement on the discrete variables, which again enables the use of a branch and bound algorithm instead of a derivative free algorithm.

Table 1 – Indices

j	-	Well
l	-	Pipeline
p	-	Phase

Table 2 – Sets

J	-	Set of wells
$J_s \subset J$	-	Set of wells connected to subsea manifold
$J_t \subset J$	-	Set of wells connected to topside manifold
L	-	Set of pipelines connected to subsea manifold
P	-	Set of phases (g for gas, o for oil and w for water)

Standard industry approach

In the following, an optimization model formulation of the "standard industry approach" is presented using the FPSO case for illustration purposes. The production system is treated as one large model and simulated as a black box, the optimization algorithm gives the simulator some input variables/parameters and the simulator calculates the output resulting variables/parameters. An objective function together with constraints on production capacity, gas lift capacity, and pressure feasibility are visible to the optimization algorithm.

The optimization loop of the standard industry approach is illustrated in Figure 7. The optimization algorithm makes a call to the production network simulator with input parameters, such as wellhead pressures, gas lift rates, on/off and routing settings, to obtain estimates for output variables/parameters such as all well rates, pipeline rates, and subsea manifold line pressures. This enables the optimization algorithm to compute the objective function value and evaluate constraint satisfaction, e.g., capacity constraints on water and gas, and pressure feasibility within the network.

Table 3 - Parameters

C_p	-	Capacity limit on mass flow of phase p to the platform
C^{GL}	-	Gas lift mass flow available for allocation
P^S	-	Inlet pressure at separator system
P_j^W	-	Maximum wellhead pressure at well j
Q_{mj}^{GL}	-	Maximum gas lift mass flow rate to well j

In this particular model, the network simulator is represented by function (4) below. The inputs, given in vector notation, are the wellhead pressures $\boldsymbol{p}^W$, the lift gas $\boldsymbol{q}^{GL}$, and the routing of all wells $\boldsymbol{x}$ and $\boldsymbol{y}$. The simulator outputs are the production mass flow rates of all phases (gas, oil and water) from all wells, $\boldsymbol{q}^W$, in addition to the pressure in the pipelines at the subsea manifold, $\boldsymbol{p}^M$. Regarding the mass energy balances in the subsea production system, i.e., the simulator also sums the production of wells that go into the same line to give the values of the pipeline flow rates, $\boldsymbol{q}^L$. Throughout the paper, all rates q are mass flow rates.

Table 4 - Variables

p_l^M	-	Pressure in line l
p_j^W	-	Wellhead pressure at well j
q_j^{GL}	-	Gas lift mass flow to well j
q_{lp}^L	-	Mass flow of phase p through pipeline l
q_{jp}^W	-	Mass flow of phase p from well j
x_j	-	1 if well j (topside manifold) is open, 0 otherwise
y_{jl}	-	1 if well j is connected to line l (subsea manifold), 0 otherwise

$$(\boldsymbol{q}^W, \boldsymbol{q}^L, \boldsymbol{p}^M) = f^{BB}(\boldsymbol{p}^W, \boldsymbol{q}^{GL}, \boldsymbol{x}, \boldsymbol{y}) \tag{4}$$

The objective function (5) seeks to maximize the oil production. It has two parts, the first part relates to the subsea wells and sums the oil flowing through each pipeline. The second part sums the oil production from each well at the topside manifold.

$$Max \sum_{l\in L} q_{lo}^L + \sum_{j\in J_t} q_{jo}^W \tag{5}$$

Water and gas production restrictions, requiring the total amount produced to be less than or equal to the gas and water handling capacity on the platform, are handled by output constraint (6). Again, the first part relates to the subsea wells, and the second to the topside wells.

$$\sum_{l\in L} q_{lp}^L + \sum_{j\in J_t} q_{jp}^W \leq C_p \qquad p \in \{g, w\} \tag{6}$$

A limited amount of gas lift is available, thus input restriction (7) requires the total amount of gas lift used to be less than or equal to this amount. Gas lift values for the topside wells, must be multiplied by the well binary variable as the gas lift rates and the binary variables are input variables and can be set independently of each other, leading to incorrect summations. A subsea well has one binary variable associated with each pipeline, this is in order to correctly route flow from the subsea wells into one of the pipelines. In this case, two binary variables are associated with each subsea well for on/off and routing.

$$\sum_{j\in J_s}\sum_{l\in L} q_j^{GL} y_{jl} + \sum_{j\in J_t} q_j^{GL} x_j \leq C^{GL} \tag{7}$$

If the binary variable associated to a certain well and line takes the value 1 it signifies that the well is routed to that line. Input constraint (8) prevents a subsea well from being routed to several lines by stating that at most one of the binary variables associated with the well can take the value 1.

$$\sum_{l\in L} y_{jl} \leq 1 \qquad j \in J_s \tag{8}$$

In order to support flow in the right direction the wellhead pressure of producing topside well must be greater than the separator inlet pressure, ensured by constraints (9).

$$P^S \leq p_j^W \qquad j \in J_t \tag{9}$$

Constraints (10) ensure that if a subsea well is connected to a line, then the wellhead pressure must be larger than or equal to the pressure in the line. The manifold pipeline pressures p_l^M are here duplicated for each well, p_{jl}^M. The restriction can then be relaxed and p_{jl}^M set to 0 when the well is not routed to line l.

$$p_{jl}^M \leq p_j^W \qquad j \in J_s \;\; l \in L \tag{10}$$

All variables take non-negative values below variable specific upper limits. Furthermore, a producing well has both upper and lower restrictions on wellhead pressure and upper and lower limits on gas lift rates.

The SmartOpt approach

Treating the network simulator as one black box and solving it with a derivative free optimization algorithm, which is the standard industry approach, does not exploit the properties and structures of the network simulators nor the variable space of the optimization problem. This is fine for "what-if" analyses, but as mentioned the approach has large limitations concerning discrete decisions and computing gradient information.

Different levels of structure exploitation is possible. In its most simple form, SmartOpt splits the optimization problem into two subspaces, one with continuous decision variables and the other with integer decision variables. A more sophisticated version, utilizes the network structure and splits the network simulator into smaller component simulators to include more information that the optimization algorithm can exploit.

Splitting the decision space

The simplest form of the SmartOpt approach splits the optimization problem into one integer subspace and one continuous subspace, as illustrated in Figure 8. The mathematical formulation remains similar to the standard approach besides this. However, such a division facilitates the use of optimization algorithms in two layers.

A derivative-free optimization algorithm may decide upon discrete variable values for all routing and on/off decisions, and act as the master algorithm. For each iteration of the derivative-free algorithm, a gradient-based optimization algorithm communicates with the network simulator, and utilizes gradient information in the search for the optimal solution of the continuous subspace.

This approach enables utilization of the available gradient information. Furthermore, the dimensions of the integer subspace are reduced to only contain the variables that truly are derivative-free.

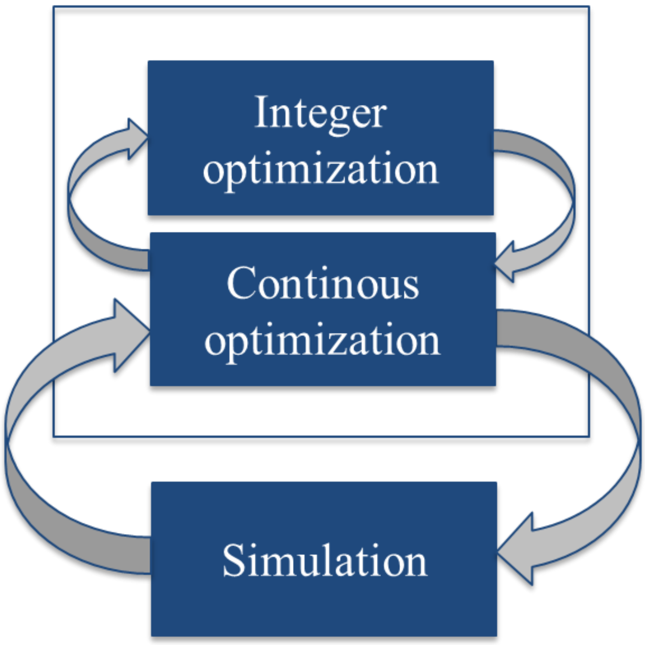

Figure 8 – the 2-layer optimization loop

Splitting up the network simulator

Figure 9 illustrates the extended SmartOpt optimization approach, which also utilizes the network structure by splitting up the network simulator into smaller component simulators. *Network* simulators , as the one shown in Figure 8 for the FPSO case, have some exploitable qualities. Mass and pressure balances through the network are easily calculated. The complex parts of the network simulator are in reality only each component, e.g., wells and pipelines, which can comprise thousands of equations and code lines. Hence, the structure of the network simulator can be formulated with very simple algebraic equations.

SmartOpt takes advantage of the network structure and splits the production network model into components, for example for each well, pipeline, compressor and separator. A distinct model and/or simulator then represents each production system component, while all mass and pressure relations within the production network are "connected" by algebraic expressions that are available to the optimization algorithm. As long as one adheres to steady-state simulation, modelling the connections as algebraic expressions is fairly straightforward (Gunnerud & Foss, 2010) (Gunnerud, et al., 2013). Another advantage is that one can call the specific component simulators as one needs information about that part of the system, which is more efficient than calling the whole network simulator each time. A necessary condition for using this modelling concept is that the system is divisible into smaller parts. The concept is visualized in Figure 11. Each black box in the figure represents one simulator giving the relations between what goes in and what comes out of that box. There are two types of simulators present in the figure, pipeline pressure drop simulators and well production simulators. Additionally, there are mass and pressure balances connecting the parts in a feasible manner.

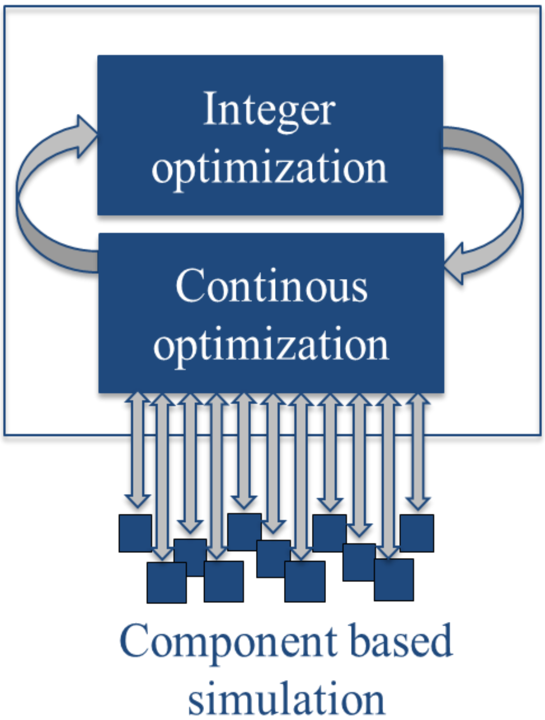

Figure 9 – SmartOpt optimization loop

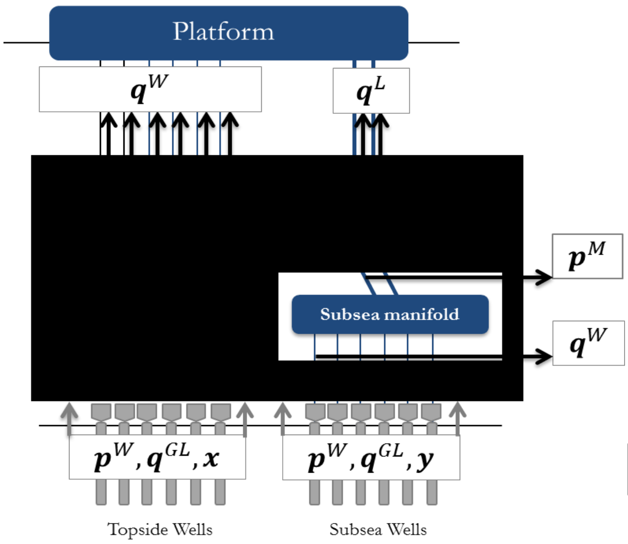

Figure 10 – Black box simulator

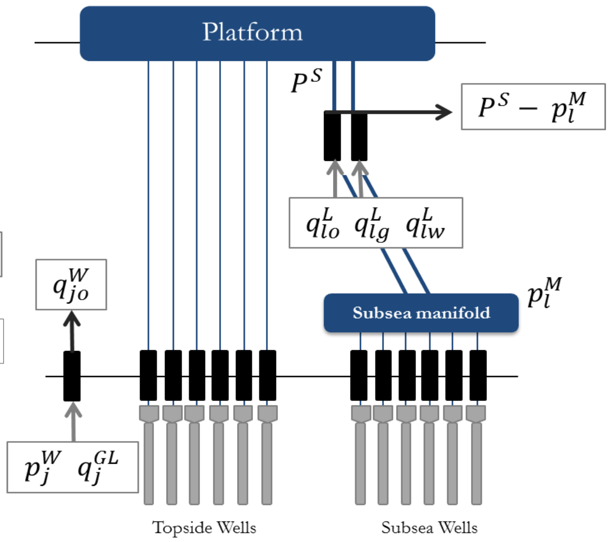

Figure 11 - Disaggregated simulators

In the above figures grey arrows depict the inputs into a simulator, while black arrows indicate the outputs. Figure 10 shows what parts of the system are covered by the black box simulator. See also equation (4) and Figure 4 for comparison. Figure 11 shows the splitting up of the one black box simulator in Figure 10 into several smaller black boxes depicting each well and each pipeline. See also equations (11) - (12).

Table 5 - Parameters

Q_{jp}^{W}	-	Maximum mass flow of phase p from well j
P_{l}^{M}	-	Maximum manifold pressure for pipeline l at subsea manifold

Table 6 - Variables

q_{jlp}^{WL}	-	Mass flow of phase p from well j routed to pipeline l at subsea manifold

The discrete routing and on/off variables, y_{jl}, are no longer contained in the simulators, they are solely associated with algebraic equations. This enables the optimization algorithm to ask for evaluations of non-discrete values on the routing and on/off variables. Thus, the branch and bound algorithm can be applied. The interested reader is referred to (Lundgren et al., 2010).

Much more information regarding the production network is now revealed to the optimization algorithm through the additional variables and constraints. This results in an optimization problem of a much higher order, thus facilitates the use of more gradient search information. That is, instead of looking at the effect on total oil production when the gas lift rate of one well is changed, one looks at the change in oil production from the well itself. The importance of accurate gradients for most optimization algorithms cannot be emphasized enough, as they essentially point the search in superior directions and lead to faster convergence of the algorithm.

In the mathematical formulation of the FPSO case SmartOpt divide the black box simulator (4) in several smaller black boxes and some connecting equations. Specifically, the separation will lead to simulators depicting well behaviors and pressure drop through the pipelines, in addition to mass and pressure balance equations making the system physically feasible.

The resulting model formulation is presented below. The production network is now split into one simulator for each well and each pipeline. Function (11) represents the well simulation, i.e. the nonlinear relationship between the wellhead pressure and gas lift rate of a well and the corresponding mass flow rates. Function (12) represents the pipeline simulation, i.e. the nonlinear relationship between the pipeline pressure drop and the oil, gas, and water that flow through it.

$$q_{jp}^{W} = f_{jp}^{W}\left(p_{j}^{W}, q_{j}^{GL}\right) \qquad p \in P \quad j \in J \tag{11}$$

$$P^{S} - p_{l}^{M} = f_{l}^{L}\left(q_{lo}^{L}, q_{lg}^{L}, q_{lw}^{L}\right) \qquad l \in L \tag{12}$$

Given below are the objective function (13) and the capacity constraints (14)-(15) for gas, water and gas lift respectively. As before, the first parts relate to the subsea wells and the second parts to the satellite wells.

$$Max \sum_{l \in L} q_{lo}^{L} + \sum_{j \in J_t} q_{jo}^{W} \tag{13}$$

$$\sum_{l \in L} q_{lp}^{L} + \sum_{j \in J_t} q_{jp}^{W} \leq C_p \qquad p \in \{g, w\} \tag{14}$$

$$\sum_{j \in J_s} \sum_{l \in L} q_{j}^{GL} + \sum_{j \in J_t} q_{j}^{GL} \leq C^{GL} \tag{15}$$

Constraints (16) prevents a subsea well from being routed to several lines by stating that at most one of the binary variables belonging to a subsea well can take the value 1.

$$\sum_{l \in L} y_{jl} \leq 1 \qquad j \in J_s \qquad (16)$$

An additional set of constraints is extracted for the mass balance equations when the black box model is disaggregated. Constraints (17)-(20) represent the mass balance formulation, and are linearly formulated based on the "Big M" parameters. (17) ensures that a satellite well is not producing when it is closed. Whenever the associated binary variable takes the values 0, the well is closed. Furthermore, if the associated binary variable takes the value 1 the restriction does not affect the problem.

$$q_{jp}^{W} \leq Q_{jp}^{W} x_j \qquad j \in J_t \quad p \in \{g, w\} \qquad (17)$$

The subsea well mass balances are given by (18)-(20), stating that the flow through a pipeline is the sum of produced rates from all subsea wells routed to that line. (19) ensures that each subsea well is only producing to the pipeline it is routed to.

$$\sum_{l \in L} q_{jlp}^{WL} = q_{jp}^{W} \qquad j \in J_s \; p \in P \qquad (18)$$

$$q_{jlp}^{WL} \leq Q_{jp}^{W} y_{jl} \qquad j \in J_s \; l \in L \, p \in P \qquad (19)$$

$$\sum_{j \in J_s} q_{jlp}^{WL} = q_{lp}^{L} \qquad l \in L \; p \in P \qquad (20)$$

Constraints (21) and (22) correspond to the pressure constraints in the black box formulation, linearly formulated based on the "Big M" method. The inequalities ensure that the solution complies with the pressure restrictions of the system.

$$p_l^M \leq p_j^W + P_l^M \, (1 - y_{jl}) \qquad j \in J_s \quad l \in L \qquad (21)$$

$$P^S \leq p_j^W + P^S (1 - x_j) \qquad j \in J_t \qquad (22)$$

5. SmartOpt approaches

Splitting up the optimization variable decision space and the network simulation model opens up for a range of alternative approaches to address the problem. The authors has studied and compared several of these. They are given names and are elaborated on in this chapter.

The first approach that naturally crystalize itself from the previous chapter, is to keep the network simulator (23) and the mathematical formulation, and divide the search space in to a derivative free master problem, and a continuous sub-problem. This approach is throughout the remaining text referred to as the *2-layer* approach.

$$(\boldsymbol{q}^W, \boldsymbol{q}^L, \boldsymbol{p}^M) = f^{BB}(\boldsymbol{p}^W, \boldsymbol{q}^{GL}, \boldsymbol{x}, \boldsymbol{y}) \qquad (23)$$

If one also splits up the network simulator as (24) and (25), many more alternatives arise. First, since the integer variables now can be relaxed, the authors has used branch and bound to handle these variables. This results in continues sub-problems which now contain both the continuous variables and the relaxed integer variables, solved within the three structure of the branch and bound algorithm. The second approach is named *SmartOpt Simulation*, and utilizing the SmartOpt mathematical formulation in the previous chapter, and calls the simulator directly to extract function value and derivatives. If the simulator do not provide derivatives, it is possible to extract these through finite differencing.

Opposite to a network simulator, which typically have more than twenty degrees of freedom, the well and pipeline simulators only have two and three respectively. This opens up the possible to sample the simulators prior to running the optimization algorithm, and instead use proxy models during the optimization itself. The authors has developed and tested several alternatives including; interpolation on the pre-sampled well and pipeline data sets, referred to as *SmartOpt interpolation*, least square fits to develop algebraic representations, referred to as *SmartOpt algebraic*, and modal reformulation of the problem to be able to use special ordered sets of type 2 (SOS2), referred to as *SmartOpt SOS2*.

$$q_{jp}^{W} = f_{jp}^{W}\left(p_j^{W}, q_j^{GL}\right) \qquad j \in J\ p \in P \tag{24}$$

$$P^{S} - p_l^{M} = f_l^{L}\left(q_{lo}^{L}, q_{lg}^{L}, q_{lw}^{L}\right) \qquad l \in L \tag{25}$$

2-layer approach

This 2-layer approach only splits up the decision space, into one discrete problem and one continuous subspace. The mathematical formulation stays similar to the black box formulation. A gradient-based algorithm solves the continuous sub problem, which should lead to faster convergence as gradient information is exploited.

This method requires the optimization algorithm to make calls to the entire network simulator, represented by (23) which is re-stated above for completeness. Thus, each time the optimization algorithm needs information about a well or a pipeline, it needs to provide all input variables, i.e. wellhead pressures, gas lift rates, on/off and routing settings, to the simulator and get information about all the output variables, i.e. well rates, pipeline rates, and subsea manifold line pressures. Derivate-free algorithms solve the discrete subspace.

SmartOpt simulation

The word "simulation" in the heading indicates that SmartOpt is solved by including the component simulators (24) and (25) directly during the optimization process.

Each well simulator provides an output of the resulting oil rate, given an input of gas lift and wellhead pressure. GOR and WC are constant for the well, thus the gas and water rates are computed algebraically. As a function of the oil, gas and water flowing through it, the pipeline simulator provides the resulting pressure drop

SmartOpt interpolation

SmartOpt interpolation utilize high-resolution tables sampled upfront to represent the well- and pipeline component simulators (11) and (12). Approximation is done by interpolating in the resulting data tables, while derivatives are obtained by finite differencing.

The interpolation code is a standalone bit of code invoked by the optimization algorithm with the necessary input values. This work includes two interpolation techniques, linear interpolation and spline interpolation, referred to as the SmartOpt linear interpolation approach and the SmartOpt cubic spline approach.

The interpolant of linear interpolation becomes a linear function between two data points. It is suitable for approximating simulators within an optimization problem due to its quick and easy nature, but the resulting function is not differentiable at the discrete points. Spline interpolation techniques choose polynomials for each interval to fit smoothly together. Splines can achieve any degree of smoothness by increasing the polynomial degree. Cubic splines are used here, the lowest degree that will produce sufficiently smooth functions (twice continuously differentiable). Splines can easily be differentiated, and their derivatives are new splines of lower degree.

SmartOpt algebraic

Table 7 – Indices

f	-	Function

Table 8 - Sets

F_{jp}^{W}	-	Set of well functions for well j phase p

Table 9 - Variables

s_{jpf}^{W}	-	1 if function f is used for well j phase p, 0 otherwise

Another interesting way to approximate a component simulator is through algebraic expressions. Creation of algebraic proxy models facilitates detection of potential non-convexities in the simulation data. Abrupt transitions between multiphase flow regimes are often the reason for such non-convexities, which can be difficult to handle and may cause the optimization algorithm to be non-convergent i.e. jump back and forth between two flow regimes. A convenient way to handle these non-convexities is by adapting proxy models for each regime, and use binary variables together with the branch and bound algorithm to decide which regime to operate within i.e. witch proxy model is active. The branch and bound algorithm makes

sure that the solution is in either one of the regimes. This may improve convergence by preventing the gradient-based solution algorithm from jumping back and forth.

The algebraic proxy models (26) - (29) represent the component simulators (24) and (25). Approximation (26) of the respective well simulator is a summation of several algebraic proxy models representing different operation regimes. To ensure that only one proxy model is "active" at any time, each proxy model has an associated binary variable multiplied to itself. Furthermore, all the binary variables that belong to a component simulator approximation must sum to 1, given by equations (27). Proxy models for the pipeline simulators are combined the same way in (28) and the binary variables summations are given in (29).

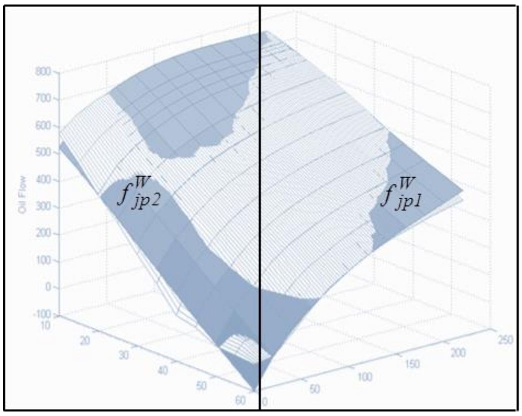

Figure 12 – Algebraic proxy model for a well simulator

Furthermore, each proxy model of a component simulator approximation is only valid for its respective production regime. Inequality constraints must be formulated in the mathematical formulation to ensure that only the correct binary variable take the values 1 for a certain regime. The number of such constraint depends on the number of regimes of the respective component simulator.

The proxy models are often nonlinear functions composed of several basis functions, e.g. polynomials. Figure 12 illustrates high-resolution data plotted together with an approximation composed of two proxy models representing their respective regimes.

$$q_{jp}^{W} = \sum_{f \in F_{jp}^{W}} f_{jpf}^{W}\left(p_j^{W}, q_j^{GL}\right) s_{jpf}^{W} \qquad j \in J\ p \in P \qquad (26)$$

$$\sum_{f \in F_{jp}^{W}} s_{jpf}^{W} = 1 \qquad j \in J\ p \in P \qquad (27)$$

$$P^{S} - p_l^{M} = \sum_{f \in F_l^{L}} f_{lf}^{L}\left(q_{lo}^{L}, q_{lg}^{L}, q_{lw}^{L}\right) s_{lf}^{L} \qquad l \in L \qquad (28)$$

$$\sum_{f \in F_l^{L}} s_{lf}^{L} = 1 \qquad l \in L \qquad (29)$$

SmartOpt SOS 2

It is possible to convert a nonlinear programming model into a suitable form for separable programming model. This can be done by constructing a piecewise linear approximation to the nonlinear model function through modal formulation and definitions of special ordered sets of type 2 (SOS2). Modal formulation with SOS 2 definitions can be used when linearizing multidimensional function, as is the case for the well (two dimensions) and pipeline (three dimensions) models. Two examples of such approximations are provided for a non-linear function of single variables and a nonlinear function of two variables. Further the linear approximations for (24) and (25) are provided.

SOS 2 definition when approximating nonlinear functions of single variables

Linear approximation of the nonlinear function $y = x^2$ with modal formulation and weighting variables defined as SOS2 is given by constraints (30) - (34). A fixed number of breakpoint values for x and y are defined and nonnegative weighting variables λ_i are assigned to each breakpoint i. Defining the weighting variables as SOS2 means that at most two points can be non-zero, and they have to be adjacent. This allows for interpolation between the two associated break points as illustrated in Figure 14. Relaxing the neighboring requirements might lead to poor function approximations as seen in the Figure 13. (30)-(33) represent the modal formulation and the inequities (34) represent the SOS2 definition. SOS 2 can also be used when linearizing multidimensional functions.

$$x = 0\lambda_1 + 1\lambda_2 + 2\lambda_3 + 2.5\lambda_3 \quad (30)$$
$$y = 0\lambda_1 + 1\lambda_2 + 4\lambda_3 + 6.25\lambda_3 \quad (31)$$

$$\lambda_1 + \lambda_2 + \lambda_3 + \lambda_3 = 1 \quad (32)$$
$$\lambda_i \in [0,1],\ i \in \{1, \dots, 4\} \quad (33)$$

$$\begin{aligned} &\alpha_1 < \lambda_1 \\ &\alpha_2 < \lambda_1 + \lambda_2 \\ &\alpha_3 < \lambda_2 + \lambda_3 \\ &\alpha_4 < \lambda_3 \\ &\alpha_i \in \{0,1\},\ i \in \{1, \dots, 4\} \end{aligned} \quad (34)$$

SOS 2 definition when approximating nonlinear functions of two variables

Linearization of the nonlinear function of x and y z= $g(x, y)$ is given by equations (35)-(42). Break points must now be chosen in two dimensions, which result in a more complicated remodeling procedure. A grid of values of (x, y) is defined with associated weighting variables λ_{sk}, illustrated in Figure 15.

$$z = \sum_{s \in S} \sum_{k \in K} g(X_s, Y_k)\lambda_{sk} \quad (35)$$

$$y = \sum_{s \in S} \sum_{k \in K} Y_k \lambda_{sk} \quad (36)$$

$$x = \sum_{s \in S} \sum_{k \in K} X_s \lambda_{sk} \quad (37)$$

$$\sum_{s \in S} \sum_{k \in K} \lambda_{sk} = 1 \quad (38)$$

$$\eta_s^S = \sum_{k \in K} \lambda_{sk} \qquad s \in S \quad (39)$$

$$\eta_k^K = \sum_{s \in S} \lambda_{sk} \qquad k \in K \quad (40)$$

$$\eta_k^K \ \textit{is SOS2 for}\ k \qquad k \in K \quad (41)$$

$$\eta_s^S \ \textit{is SOS2 for}\ s \qquad s \in S \quad (42)$$

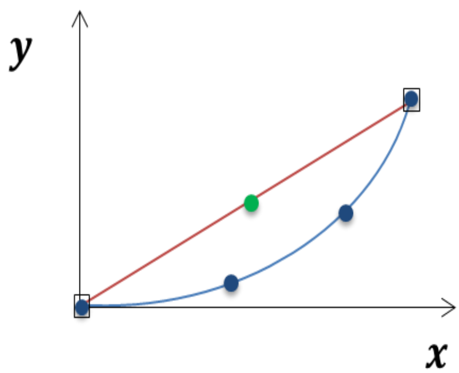

Figure 13 – SOS 2 formulation with relaxed neighboring requirements

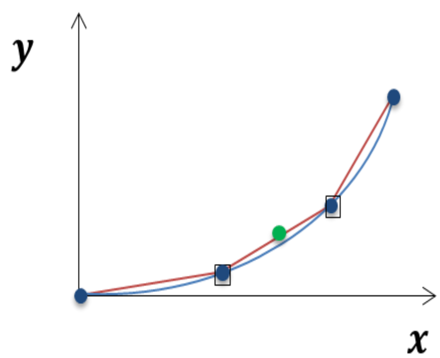

Figure 14 – SOS 2 formulation with neighboring requirements

The values (x, y) at the grid points are denoted (X_s, Y_k). Four neighboring weighting variables can be non-zero, because there are break points in two dimensions. This condition is a generalization of a SOS2 set, and is imposed by equations (39)-(42). Since a SOS2 variable set needs to be a one dimensional vector, η_s^S and η_k^K are introduced as auxiliary weighting variables of S and K. The SOS2 conditions (41) for set S allows λ_{sk} to be non-zero in at most two neighbouring rows. Similarly, the SOS2 conditions (42) for set K allows λ_{sk} to be non-zero in at most two neighbouring columns. Together, they enforce that only four adjacent λ_{sk} weighting variables are non-zero.

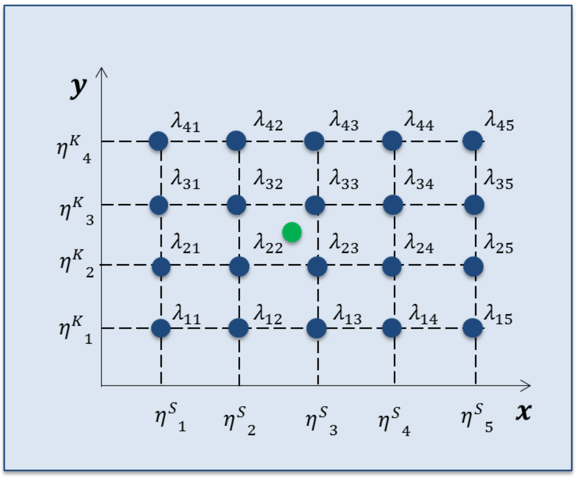

Figure 15 – Grid of values for (x,y) with associated weighting variables λ

SOS2 well approximation

As the well flow is dependent on wellhead pressure and gas lift, breakpoints must be chosen in two dimensions and a surface (q_{jo}^W, q_j^{GL}) can be defined. Associated to each point on this surface there is a weighting variable $\lambda_{(j)kn}$. Breakpoint k gives a corresponding wellhead pressure, $P_{(j)k}^W$ and breakpoint n gives a corresponding gas lift flow rate, $Q_{(j)n}^{GL}$. $Q_{(jo)kn}^W$ is the oil flow rate corresponding to breakpoint kn. The oil flow rate for well j connected can now be approximated by the following equations (43)-(54).

Some of the production wells have semi-continuous production curves, which mean that the wells will produce only when the associated gas lift rate is above a certain limit. These properties are modelled by (53) and (54).

Table 10 - Indices

k	-	Breakpoint
n	-	Breakpoint

Table 11 - Sets

K	-	Set of breakpoints for values of p_{mj}^W
N	-	Set of breakpoints for values of q_{mj}^I

Table 12 - Parameters

$P_{(j)k}^W$	-	Value of p_j^W at breakpoint k
$Q_{(j)n}^{GL}$	-	Value of q_j^{GL} at breakpoint n
$Q_{(jo)kn}^W$	-	Value of q_{jo}^W at breakpoints k and n

Table 13 - Variables

$\lambda_{(j)kn}$	-	Weight of breakpoint k and n for well j
$\eta_{(j)k}^{KW}$	-	Sum of weights of breakpoints n for breakpoint k and well j
$\eta_{(j)n}^{NW}$	-	Sum of weights of breakpoints k for breakpoint n and well j

$$p_j^W = \sum_{k \epsilon K} \sum_{n \epsilon N} P_{(j)k}^W \lambda_{(j)kn} \qquad j \epsilon J \qquad (43)$$

$$q_j^{GL} = \sum_{k \epsilon K} \sum_{n \epsilon N} Q_{(j)n}^{GL} \lambda_{(j)kn} \qquad j \epsilon J \qquad (44)$$

$$q_j^{GL} = \sum_{k\epsilon K}\sum_{n\epsilon N} Q_{(j)n}^{GL}\lambda_{(j)kn} \qquad j \in J \tag{45}$$

$$\sum_{k\epsilon K}\sum_{n\epsilon N} \lambda_{(j)kn} = 1 \qquad j \in J \tag{46}$$

$$\lambda_{(j)kn} \geq 0 \qquad j \in J \quad k \in K \quad n \in N \tag{47}$$

$$\eta_{(j)k}^{K} = \sum_{n\epsilon N} \lambda_{(j)kn} \qquad j \in J \quad k \in K \tag{48}$$

$$\eta_{(j)n}^{N} = \sum_{k\epsilon K} \lambda_{(j)kn} \qquad j \in J \quad n \in N \tag{49}$$

$$\eta_{(j)k}^{K} \qquad is\ SOS\ 2\ for\ k \qquad j \in J \quad k \in K \tag{50}$$

$$\eta_{(j)n}^{N} \qquad is\ SOS\ 2\ for\ n \qquad j \in J \quad n \in N \tag{51}$$

$$\eta_{(j)n}^{N}, \eta_{(j)k}^{K} \geq 0 \qquad j \in J \quad k \in K \quad n \in N \tag{52}$$

$$\lambda_{(j)k1} \leq 1 - \sum_{l\epsilon L} y_{jl} \qquad j\epsilon J_s \quad k \in K \tag{53}$$

$$\lambda_{(j)k1} \leq 1 - x_j \qquad j\epsilon J_t \quad k \in K \tag{54}$$

SOS2 pipeline approximation

To approximate pressure drop through each pipeline l connected to the subsea manifold by the use of SOS2 formulations, breakpoints must be chosen from three dimensions. A grid $(q_{lo}^L, q_{lg}^L, q_{wo}^L)$ can be defined given by the variables for oil, gas and water flow. For each point in the grid, there is associated a weighting variable $\delta_{(l)knr}$ and values for gas, oil and water. $Q_{(l)k}^{O}$ is the oil flow value at breakpoint k, $Q_{(l)n}^{G}$ is the gas flow value at breakpoint n and $Q_{(l)r}^{W}$ is the water flow value at breakpoint r. Associated with the breakpoints k, n and r are the values for the pressure drop, $F_{(l)knr}^{L}$. The pressure drop can then be approximated for each pipeline l by the following equations (55) - (60).

The formulation will become as below in equations (61) - (67), where $\gamma_{(l)r}^{R}$, $\gamma_{(l)n}^{N}$, and $\gamma_{(l)k}^{K}$ are defined as SOS 2 sets.

Table 14 - Indices

k	-	Breakpoint
n	-	Breakpoint
r	-	Breakpoint

Table 15 - Sets

K	-	Set of breakpoints for values of q_{lo}^L
N	-	Set of breakpoints for values of q_{lg}^L
R	-	Set of breakpoints for values of q_{lw}^L

Table 16 - Parameters

$Q_{(l)k}^{O}$	-	Value of q_{lo}^L at breakpoint k
$Q_{(l)n}^{G}$	-	Value of q_{lg}^L at breakpoint n
$Q_{(l)r}^{W}$	-	Value of q_{lw}^L at breakpoint r
$F_{(l)knr}^{L}$	-	Value of f_l^L at breakpoint k, n and r

Table 17 - Variables

$\delta_{(l)knr}$	-	Weight of breakpoint k, n and r for pipeline l at the subsea manifold
$\gamma_{(l)k}^{K}$	-	Sum of weights of breakpoint n and r for breakpoint k for pipeline l at the subsea manifold
$\gamma_{(l)n}^{N}$	-	Sum of weights of breakpoint k and r for breakpoint n for pipeline l at the subsea manifold
$\gamma_{(l)r}^{R}$	-	Sum of weights of breakpoint k and n for breakpoint r for pipeline l at the subsea manifold

$$q^L_{lo} = \sum_{k\epsilon K}\sum_{n\epsilon N}\sum_{r\epsilon R} Q^O_{(l)k}\delta_{(l)knr} \qquad l \epsilon L \qquad (55)$$

$$q^L_{lg} = \sum_{k\epsilon K}\sum_{n\epsilon N}\sum_{r\epsilon R} Q^G_{(l)n}\delta_{(l)knr} \qquad l \epsilon L \qquad (56)$$

$$q^L_{lo} = \sum_{k\epsilon K}\sum_{n\epsilon N}\sum_{r\epsilon R} Q^W_{(l)r}\delta_{(l)knr} \qquad l \epsilon L \qquad (57)$$

$$\sum_{k\epsilon K}\sum_{n\epsilon N}\sum_{r\epsilon R} F^L_{(l)knr}\delta_{(l)knr} = P^S - p^M_l \qquad l \epsilon L \qquad (58)$$

$$\sum_{k\epsilon K}\sum_{n\epsilon N}\sum_{r\epsilon R} \delta_{(l)knr} = 1 \qquad l \epsilon L \qquad (59)$$

$$\delta_{(l)knr} \geq 0 \qquad l \epsilon L \quad k \epsilon K \quad n \epsilon N \quad r \epsilon R \qquad (60)$$

$$\gamma^R_{(l)r} = \sum_{k\epsilon K}\sum_{n\epsilon N} \delta_{(l)knr} \qquad l \epsilon L \quad r \epsilon R \qquad (61)$$

$$\gamma^N_{(l)n} = \sum_{k\epsilon K}\sum_{r\epsilon R} \delta_{(l)knr} \qquad l \epsilon L \quad n \epsilon N \qquad (62)$$

$$\gamma^K_{(l)k} = \sum_{n\epsilon N}\sum_{r\epsilon R} \delta_{(l)knr} \qquad l \epsilon L \quad k \epsilon K \qquad (63)$$

$$\gamma^R_{(l)r} \ is\ SOS\ 2\ for\ r \qquad l \epsilon L \quad r \epsilon R \qquad (64)$$

$$\gamma^N_{(l)n} \ is\ SOS\ 2\ for\ n \qquad l \epsilon L \quad n \epsilon N \qquad (65)$$

$$\gamma^K_{(l)k} \ is\ SOS\ 2\ for\ k \qquad l \epsilon L \quad k \epsilon K \qquad (66)$$

$$\gamma^R_{(l)r}, \gamma^N_{(l)n}, \gamma^K_{(l)k} \geq 0 \qquad l \epsilon L \quad k \epsilon K \quad n \epsilon N \quad r \epsilon R \qquad (67)$$

6. Implementation

In the sequel we present seven different ways of implementing the optimization formulations. These are summarized in the top line of Table 18 and include the standard (black box) approach and six different strategies for implementing SmartOpt. This is presented in some detail below. The six SmartOpt approaches differ through the choice of component models, which also has a bearing on the chosen optimization algorithm.

If simulators are included directly in the optimization algorithm or approximated by a stand-alone interpolation code, it is not possible to use state-of-the-art modelling software. This forces the user to implement the model formulation in c++ and solve it using a proper optimization algorithm.

The black box approach is solved by the state-of-the-art optimization solver called NOMAD which implements a derivative free optimization algorithm called Mesh adaptive direct search (MADS). For more information about MADS see (Digabel, 2011). The 2-layer technique applies NOMAD on top to decide on the integers and IPOPT (COIN-OR, Ipopt website, 2012), i.e., an interior point methods for the continuous variables. For each integer solution, the gradient-based optimization algorithm IPOPT solves the continuous optimization problem. Both the first and second order derivatives are computed numerically by finite differencing and the quasi-Newton BFGS method.

Bonmin solves SmartOpt simulation and the two SmartOpt interpolation methods. Bonmin is an open source optimization solver within the COIN-OR framework (COIN-OR, Bonmin website, 2012). It is a general MINLP solver and as such fits well to the models given here. Bonmin uses IPOPT to solve the relaxed NLPs, while a branch and bound option exists for handling discrete variables. The first and second order derivatives are computed numerically by finite differencing and the quasi-Newton BFGS method.

SmartOpt algebraic and SmartOpt SOS 2 can both be implemented in state-of-the-art modelling languages, AMPL and Mosel respectively. SmartOpt algebraic are further solved by Bonmin. SmartOpt SOS2 are solved by Xpress, a solver developed specifically in combination with Mosel. Xpress Mosel utilize a branch and bound algorithm in order to handle discrete decisions and simplex to solve the continuous parts. Xpress Mosel is limited to solve LPs and MILPs.

Table 18 – Implementation strategies

<table>
<tr><th rowspan="2"></th><th rowspan="2">Black box</th><th colspan="6">SmartOpt</th></tr>
<tr><th>2-layer</th><th>Sim.</th><th>Linear int.</th><th>Cubic spline</th><th>Algebraic</th><th>SOS2</th></tr>
<tr><td>Implementation language</td><td colspan="5">C++</td><td>AMPL</td><td>MOSEL</td></tr>
<tr><td>Optimization problem class</td><td colspan="3">MINLP “simulation”</td><td colspan="3">MINLP</td><td>MILP</td></tr>
<tr><td>Algorithm</td><td>MADS</td><td>MADS + Interior point</td><td colspan="4">B&B + Interior point</td><td>B&B + Simplex</td></tr>
</table>

7. Computational study and results

Given the capacity for handling gas and water, available gas lift and constant separator pressure, each solution approach should find the optimal wellhead pressures and lift gas allocations, as well as routing of subsea wells to lines. Six cases are defined for available gas lift, gas and water handling capacities and separator pressure in order to compare the results of the solution algorithms and to get a sense of the robustness. The variations in these parameters will affect the production rates differently, e.g. increased capacity limits allow for more production. An increase in separator inlet pressure will have the opposite effect, the pressure difference between the wellhead pressures and the platform pressure will be reduced and less fluid can be carried up from the reservoir.

Whereas the nonlinear formulations are dependent on an initial solution, and may only provide local optimal solutions, the solutions resulting from SmartOpt SOS2 are considered the global optima. Thus, the SOS2 results are considered optimality benchmarks to evaluate the solution quality of the other methods.
Apart from SmartOpt SOS2, all approaches are run with six different initial solutions generated for this

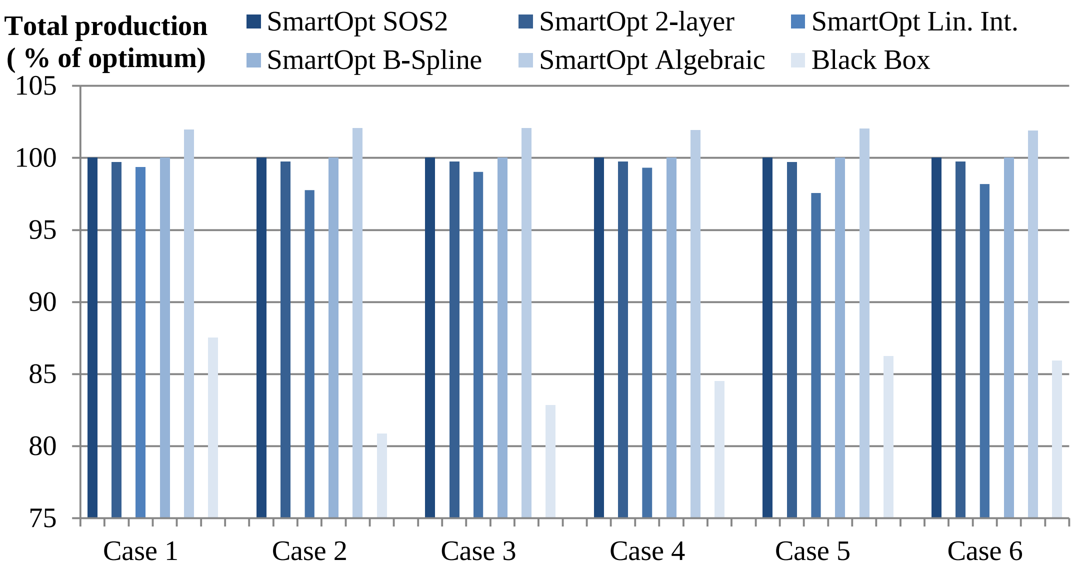

Figure 16 – Case 1 production rates

purpose. Figure 16 illustrates production rates provided by the solution methods in total oil production rates extracted from the system.

Table 19 – Best objective functions and variations given in percentage of SOS2 solution

		SmartOpt				
	Black box	**2-layer**	**Linear int.**	**Cubic spline**	**Algebraic**	**SOS2**
Case 1	84.7± 9.1 %	99.7±18.4 %	99.4± 6.4 %	100.0±0.0 %	102.0±0.3 %	100 %
Case 2	95.0±21.5 %	99.7±18.7 %	97.8±13.1 %	100.0±0.0 %	102.1±0.2 %	100 %
Case 3	89.5±15.3 %	99.7±18.9 %	99.0± 7.3 %	100.1±0.0 %	102.1±0.0 %	100 %
Case 4	92.5±18.3 %	99.8±18.3 %	99.3± 7.8 %	100.0±0.0 %	101.9±0.0 %	100 %
Case 5	88.0±10.9 %	99.7±18.5 %	97.6± 3.2 %	100.1±0.0 %	102.0±0.1 %	100 %
Case 6	92.3±17.8 %	99.7±18.0%	98.2± 9.6 %	100.1±0.0 %	101.9±0.0 %	100 %

Table 20 - Best solution times for Case 1

		SmartOpt				
	Black box	**2-layer**	**Linear int.**	**Cubic spline**	**Algebraic**	**SOS2**
Time (sec)	237	151	0.72	107	14	463

The best objective function for each case and the variations are provided in the table 19. The variations give a measure of the robustness of the solution methods, and are defined as the difference between the best and worst objective function values for each case. All values are given as percentage relative to the SOS2 solutions for ease of comparison.

In table 20, the best solution times for case 1 are listed only to provide the reader with an approximate measure on the efficiency of the different approaches. However, keep in mind that the methods are run with both different software and hardware. Thus, in reality the times presented in table 20 cannot be directly compared.

Uniquely different, the methods presented are in truth all only approximations of the reality. For unbiased comparison of the methods, the black box model is used to evaluate the feasibility or possibly infeasibility of the solutions. The SmartOpt cubic spline solutions prove feasible (sometimes breaking a few pressure balances by less than 3×10^{-4} %). To a varying degree, all other solutions are deemed infeasible. SmartOpt-2 layer, SmartOpt linear interpolation and SmartOpt SOS2 solutions are only slightly infeasible, typically breaking the pressure balance restrictions for one or two wells by less than ~0.3 %. The SmartOpt Algebraic approach solutions however, break the pressure restrictions by almost 3 %. The reason for these infeasibilities is the component simulations approximations. They are generated with the least squares method, which means that the some well approximation might end up overestimating the oil production for certain wellhead pressures and gas lift allocations, while pipeline approximations at the same time underestimate the pressure losses over the pipelines for the resulting flows.

8. Discussion

There are many applications where simulators are structured in networks, and where these network simulators easily can be broken down into smaller stand-alone components. This is particularly true for process simulators. However, the applicability of this optimization approach is quite general. The SmartOpt optimization approach is tested on a realistic petroleum production problem, where the computational study shows a significant improvement for the concept compared to a standard black box non-invasive approach. The solution quality in terms of oil production is improved by more than 18 %. This is a very large number in this context, and will most probably not hold in all applications. Table 21 gives an overview of the pros and cons of all presented solution approaches.

Discrete decisions

To treat discrete decisions in optimization algorithms is challenging and can be both computationally expensive and lead to non-convergence. When the production system network is simulated a black box (4), many simulators only allow the optimization algorithm to make queries with discrete values on the routing and on/off decisions. This excludes the use of Branch and Bound based algorithms that rely on relaxing the integer requirements while solving the sub problems. Gradient information can be attained from the network simulator, however not for discrete decision variables. This further enforces a necessity for derivative-free optimization algorithms such as Genetic algorithms (GA), Generalized pattern search (GPS) or Mesh adaptive direct search (MADS) (Conn, et al., 2009).

The 2-layer approach splits the solution space into one integer and one continuous subspace. This facilitates better utilization of the available gradient information on the continuous variables, and use of continuous solvers. Furthermore, the dimensions of the integer subspace are reduced to only contain the variables that are truly derivative free. The 2-layer approach reduces the convergence issues of the standard industry approach, which as can be seen in table 19 affect both solution time and quality positively.

For the reaming SmartOpt approaches, the discrete routing and on/off variables are no longer contained in the simulators, they are solely associated with algebraic equations. This advantage enables the optimization algorithm to ask for evaluations of non-discrete values on the routing and on/off variables. Furthermore, much more information regarding the production network is revealed to the optimization algorithm, thus facilitates the use of more gradient information. The problems can then be solved by gradient-based algorithms together with branch and bound, which resulted in a beneficial effect on convergence and solution time.

Component simulators

Rather than the complete production network, several of the SmartOpt alternatives use single well/pipeline simulators or proxy's to estimate first and second order gradients. Thus, another reason for faster convergence is the fact that the simulator search are less time demanding due to fewer dimensions in the component simulators. An estimate of what will happen to the well flow rate if the well head pressure is changed is more predictable than the consequence for the complete production network if one changes the same well head pressure, or if one reroutes a well to another pipeline.

Furthermore, since the dimensions of the component simulators are low, e.g. two and three for the wells and pipelines respectively, it is possible to sample the search space upfront and create proxy models. This is not possible for the non-invasive approach, as search space usually is too large, as in this case with 24 continuous variables.

However, there are some significant challenges related to the SmartOpt approach. It is a more complex task to divide the network simulator into components, and to update high-resolution tables and/or proxy models for each of them, compared to a single non-invasive implementation. In the non-invasive case where a derivative free algorithm is used, it is more and less straightforward to implement an optimization algorithm on top of the simulator. For the proposed SmartOpt approach, there is a need for in-depth problem, simulator and optimization knowledge. The presented approach is also limited to network simulators that are easily decomposed, meaning that applicability to dynamic simulators where components are closely interconnected through dynamic behavior is still an open research question.

Model exactness

Attention should be paid to the fundamental differences of the SmartOpt simulator approximations. The number of data points and the interpolation schemes affects the accuracy of SmartOpt interpolation. Based on high-resolution tables and interpolation, both SmartOpt interpolation approaches provide realistic portrayals of the production network. The fast solution time and satisfactory accuracy of the interpolating codes might therefore undermine inclusion of the simulators directly, which is associated with considerable interfacing issues and slowing down of the algorithm.

A SOS2 formulation is highly dependent on the resolution in the data tables. Both acceptable accuracy and satisfactory solution times are obtained here. A more detailed discretization lead to a better match with the realities of the problem, but also adversely affects solution time as the number of variables increase rapidly.

Algebraic proxy models have been created based on human visual inspections combined with least square fits, which is discretionary at best. Solutions are proven slightly infeasible, undermining the accuracy of the approximations used.

Solution time and scalability

SmartOpt linear interpolation approach proves fast, providing results in less than a second. The SmartOpt algebraic approach provides solutions within a few seconds, whereas the SmartOpt cubic spline and the 2-layer approaches provide solutions within a couple of minutes. SmartOpt SOS2 and black box are rather slow and solve the base case after 463 seconds and 237 seconds respectively.

Thus, the fastest SmartOpt solution times sees a significant speed-up of up to two orders of magnitude compared to industry standard approach. This is without exploring the option of solving the component simulators in parallel. The black box derivative free algorithm is also easily parallelized, but it cannot be parallelized on a component level. This tells us, not surprisingly, that much is gained by including explicit structural constraints as much as possible. The time it takes for an algorithm to output results might affect its applicability. No doubt low solution time is greatly appreciated, and in some situations crucial. Fast algorithms encourage real-time use, and can be a useful addition to decision support systems. Due to implementations issues the SmartOpt simulation approach, including simulators directly into the optimization, is not tested. The presumption is that this modification will negatively affect solution time.

It is straightforward to scale the problem formulations up to 30+ wells. However, it becomes more challenging to solve the optimization problem. Some of the methods will see the solution speed decrease extensively with an increase in network components, as pipes and wells. The increased number of decision variables that follows a larger production network is likely to slow down the derivative-free black box approach. Evaluations of more proxy models will have a noticeable effect on the solution speed of SmartOpt algebraic. The SOS2 formulation will also be affected as the number of variables increases rapidly with more SOS2 formulations. SmartOpt SOS2 is designed to solve the global optimization problem, really only suitable for relatively low dimensional problems and would certainly be inefficient when local optima suffice.

SmartOpt interpolations methods scale better as the interpolations are quick, more interpolation-simulations must be run, but with a minimal effect on total solution speed.

Tables of higher resolutions affect the SOS2 formulation approach significantly, as the number of variables increase rapidly with the amount of data points. All remaining methods scale better in such situations, as more data points hardly affect the simulator approximation evaluations and solution speed. If exploring the option of solving the component simulator in parallel, solution times should see a speed up. This applies for all SmartOpt methods.

Solution quality
Only the SOS2 formulation guarantees a global solution. The solution will lay in-between the points of the data tables on linear segments generated by the SOS2 formulation, and the resolution will clearly affect the accuracy of this optimal solution relative to the "real" optimal solution. The SOS2 solutions are used as benchmarks in the remaining discussions.

When only local optima are possible, competing algorithms are likely to converge to different solutions. In this computational study, the interpolation approaches performs well. Cubic spline approximations prove superior and provide the optimal solutions for all cases. SmartOpt linear interpolation obtain solutions that averages on 99% of the global optima. Maybe more surprisingly, the 2-layer approach provides some solutions for each that are slightly closer to the global optima. Clearly sensitive to the exactness of the proxy models each solution of the SmartOpt algebraic approach overestimates production and surpasses the optimal objective function value. The black box optimization performs consistently worse, where the best solution averages on 90% of global optima.

Robustness
Robustness is an important issue when analyzing optimization solution techniques. Since the underlying problem is non-convex, solutions obtained by local search algorithms cannot be taken as guaranteed global optimal solutions. As the name indicates, local solutions will be found with no quality measure. Cubic spline approximations provide solutions with no variation across starting points, proving to be highly reliable and robust. Even algebraic proxy models provide an average variation across all cases for all starting points of insignificant 0.11 %. However, robustness is the biggest challenge of linear interpolation approximations, which provide solutions with an average variation of 8 %. It is highly recommended that this algorithm is run with a suitable number of start solutions, found through random generation or more sophisticated strategies. Since this method at its current form takes less than a second to solve, solving the problem for numerous starting points is not a hinder. 2-layer- and black box optimization seem particularly sensitive to initial solutions, with variations that averages at 18 % and 15 % respectively. Long solution times makes running the two latter models for a large amount of starting points less attractive.

Robustness considering the physical properties of the system is also an important issue. Situations may occur where there are abrupt changes in production data, e.g. pipelines may experience sudden changes in pressure loss due to changes in flow regimes. A valuable quality of algebraic proxy models is the sophisticated handling of such situation. Convergence is ensured through the division of space and creation of two functions approximating each distinct part of the underlying nonlinearity, where binary variables are handled through branch and bound. SmartOpt cubic spline also handles such situations very elegantly, spline interpolation techniques smoothen out the transitions and one avoids jumps in the function derivatives.

The opposite is true for the SmartOpt linear interpolation, which struggles in the situation described above. Linear interpolation is not very precise and the resulting function is not differentiable at the discrete points. This can cause problems for a numerical solver, which is highly dependent on the direction given by the derivatives. Discontinuous derivatives can cause a loop of hopping back and forth between the two distinct parts and hinder convergence.

Modelling effort

SOS 2 formulations are widely used in optimization within several industries, some optimization software (Xpress Mosel is an example) even include specific SOS2 handling, making the implementation straightforward. For the other approaches, interfacing relevant simulators into the optimization algorithm, or creating the mimicking interpolation codes and develop suitable schemes for acquiring derivative information poses implementation challenges. Generation of the proxy models on the other hand, is time consuming. Data tables of more than three to four dimensions are particularly challenging, as visualization becomes difficult. Trade-off between model quality and development time becomes a key issue.

Automation

The frequency of changing data depends on the specific asset and corresponding reservoir dynamics. Automatic fine-tuning of the simulators after new information is available can be taken as given. Including automatic updating of the data files is an easy task, warranting that all the solution methods studied here are well suited for and can handle such updated information.

Table 21 – Discussion summary

		SmartOpt				
	Black box	**2-layer**	**Linear int.**	**Cubic spline**	**Algebraic**	**SOS2**
Discrete decisions	÷	+	+ +	+ +	+ +	+ +
Component simulators	÷	÷	+ +	+ +	+ +	+ +
Model exactness	----------	---------	+	+ +	÷	+
Solution time and scalability	÷	÷	+ +	+ +	+	÷
Solution quality	+	+	+ +	+ +	+	+ +
Robustness	÷	÷	÷	+ +	+ +	+ +
Modelling effort	+	+	+	+	÷	+ +
Automation	+ +	+ +	+ +	+ +	+	+ +

Furthermore, new equipment or components might be installed on the asset. This can be handled in all the models, although they will require varying amount of effort. Updating and generation of proxy models will require some effort if new well and pipeline components are installed. The other methods will only need updating of the structure of the model, by including the additional component and taking care to add its contribution into the relevant equations. In conclusion, all methods are suited for automatic updating and fit for being included in decision support systems.

9. Conclusion

In this paper we present a group of algorithms that explores the natural structure of a network simulator, referred to as SmartOpt approaches, to find the best possible simulator settings given a predefined objective. The methods are demonstrated on a realistic petroleum production problem and show speed-ups of up to two orders of magnitude, compared to a standard approach. The solution quality is also improved.

Black box optimization is clearly inferior in terms of all standards except simplicity and ease of implementation. It does not facilitate a relaxation of integer variables, and the formulation requires a derivative free search algorithm, which often leads to long solution times or non-convergence. This leads to concluding that better solution methods are available and the authors has developed some of those.

In its most simple form, SmartOpt splits the optimization problem into two subspaces, one continuous and one integer. Such division facilitates the use of optimization algorithms in two layers, which speeds-up the solution times and improves the solution qualities. The more sophisticated SmartOpt approach also divide the one black box simulator into several black boxes depicting smaller and divisible parts of the system. Simulators are approximated and included in the algorithms as high-resolutions tables and interpolations schemes, algebraic proxy models or piecewise linearization approximations through SOS2 formulations. The latter guarantees global optimal solutions to the linear approximations performed; the formulation is robust and provides high quality solutions. Proxy models tend to cause infeasible solutions and require a significant amount of pre-solving effort. Simulators approximated through high resolution tables and interpolation result in high quality solutions. Cubic spline approximations appears independent of initial starting point and provide the optimal solution within just over a minute. Linear interpolation solve within in less than a second, although the solutions are inferior and it is fairly reliant on initial solution point.

The authors believe the advantages of the SmartOpt approach will improve further, as the size of the network grows. Potentially, it should be possible to handle problem instances with 10–100 times more optimization variables, than what is possible today.

Bibliography

Almklov, P. & Gunnerud, V., 2013. *Dailiy Production Optimization.* 1 ed. White Paper, Trondheim: Center for Integrated Operations in the Petroleum Industry.

Alpha Thames Ltd, 2004. *Dictionary of Oil Industry Terminology.* [Online]
Available at: http://www.scribd.com/doc/19552139/Dictionary-of-Oil-Industry-Terminology

AMPL Optimization LLC, 2012. *AMPL website.* [Online]
Available at: http://www.ampl.com/
[Accessed 15 12 2012].

COIN-OR, Bonmin website, 2012. [Online]
Available at: https://projects.coin-or.org/Bonmin
[Accessed 15 12 2012].

COIN-OR, Ipopt website, 2012. [Online]
Available at: https://projects.coin-or.org/Ipopt
[Accessed 15 12 2012].

Conn, A., Scheinberg, K. & Vicente, L., 2009. *Introduction to Derivative-Free Optimization. MPS/SIAM Book Series on Optimization.*. 1 ed. Philadelphia: SIAM.

Digabel, S., 2011. NOMAD: Nonlinear Optimization with the MADS algorithm. *ACM Transactions on Mathematical Software (TOMS).*

Dür, M., 2001. Dual bounding procedures lead to convergent branch-and-bound algorithms. *Mathematical Programming Series A,* Volume 91, pp. 117-125.

Echeverría Ciaurri, D., Isebor, O. J. & Durlofsky, L. J., n.d. Application of derivative-free methodologies to generally constrained oil production optimisation. *International Journal of Mathematical Modelling and Numerical Optimisation,* Volume 2, pp. 134-161.

Echeverría Ciaurri, D., Mukerji, T. & Durlofsky, L. J., 2011. *Derivative-free optimization for oil field operations.* In X.-S. Yang, & S. Koziel ed. Berlin Heidelberg:: Springer-Verlag.

Foss, B., 2012. Process control in conventional oil and gas field – Challenges and opportunities. *Control Engineering Practice,* Volume 20, pp. 1058-1064.

Fu, M., 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal of Computing,* pp. 192-215.

Gunnerud, V., Conn, A. & Foss, B., 2013. Embedding Structural Information in Simulation-based Optimization. *To appear in Computer and Chemical Engineering,* 53(1), pp. 35-43.

Gunnerud, V. & Foss, B., 2010. Oil production optimization—A piecewise linear model, solved with two decomposition strategies. *Computers and Chemical Engineering,* 34(11), pp. 1803-1812.

Gunnerud, V. & Langvik, E., 2007. *Production Planning Optimization for the Troll C field,* Trondheim, Norway: s.n.

Jakobsson, S. P. M., Rudholm, J. & Wojciechowski, A., 2009. A method for simulation based optimization using radial basis functions. *Optimization and Engineering,* Volume 11.

Jaluria, Y., 2009. Simulation-based optimization of thermal systems. *Applied Thermal Engineering,* Volume 29, pp. 1346-1355.

Jansen, J. D., Bosgra, O. H. & Van den Hof, P. M. J., 2008. Model-based control of multiphase flow in subsurface oil reservoirs. *Journal of Process Control,* 18(9), pp. 846-855.

Lundgren, J., Värbrand, P. & Rönnqvist, M., 2010. *Optimization.* 1 ed. Lund: Studentlitteratur AB.

Naevdal, G., Brouwer, D. R. & Jansen, J. D., 2006. Waterflooding using closed-loop control. *Computational Geosciences,* Volume 10, pp. 37-60.

Schwartz, J. D., Wenlin, W. & Rivera, D. E., 2006. Simulation-based optimization of process control policies for inventory management in supply chains. *Automatica,* Volume 42, pp. 1311-1320.

Takacs, G., 2005. *Gas Lift Manual.* 1st ed. Miskolc, Hungary: Pennwell Books.

Tuy, H., 2005. On solving nonconvex optimization problems by reducing the duality gap. *Journal of Global Optimization,* Volume 32, pp. 349-365.

Ursin-Holm, S. et al., 2014. *Decision Support for Daily Production Optimization: In Genuine Real-Time.* Utrecht, The Netherlands, Society of Petroleum Engineers.

Wächter, A., 2009. *Short Tutorial: Getting Started With Ipopt in 90 Minutes.* [Online]
Available at:
http://drops.dagstuhl.de/volltexte/2009/2089/pdf/09061.WaechterAndreas.Paper.2089.pdf

Wan, X., Pekny, J. F. & Reklaitis, G. V., 2005. Simulation based optimization with surrogate models-application to supply chain management. *Computers and Chemical Engineering,* Volume 29, pp. 1317-1328.

Wikipedia, 2012. *Bilinear Interpolation.* [Online]
Available at: http://en.wikipedia.org/wiki/Bilinear_interpolation
[Accessed 15 12 2012].

Wikipedia, 2012. *Trilinear Interpolation.* [Online]
Available at: http://en.wikipedia.org/wiki/Trilinear_interpolation
[Accessed 12 10 2012].