Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Norwegian University of Science and Technology, Trondheim, Norway, May 31 - June 1, 2012

Optimization of a Simulated Well Cluster using Surrogate Models ⋆

Bjarne Grimstad * **Håvard Ausen** ** **Victoria Lervik** *** **Vidar Gunnerud** **** **Dag Ljungquist** †

* *Department for Engineering Cybernetics, NTNU (e-mail: bjarne.grimstad@itk.ntnu.no)*
** *Department for Engineering Cybernetics, NTNU (e-mail: hararau@gmail.com)*
*** *Department of Industrial Economics and Technology Management, NTNU (e-mail: victoria.lervik@gmail.com)*
**** *Department for Engineering Cybernetics, NTNU (e-mail: vidargu@ntnu.no)*
† *FMC Technologies, Asker, Norway (e-mail: dag.ljunquist@fmcti.com)*

Abstract: In this paper we present an implementation of a partly Derivative-Free Optimization (DFO) algorithm for production optimization of a simulated multi-phase flow network. The network consists of well and pipeline simulators, considered to be black-box models without available gradients. The algorithm utilizes local approximations as surrogate models for the complex simulators. A Mixed Integer Nonlinear Programming (MINLP) problem is built from the surrogate models and the known structure of the flow network. The core of the algorithm is IBM's MINLP solver Bonmin, which is run iteratively to solve optimization problems cast in terms of surrogate models. At each iteration the surrogate models are updated to fit local data points from the simulators. The algorithm is tested on an artificial subsea network modeled in FlowManager$^{\text{TM}}$, a multi-phase flow simulator from FMC Technologies. The results for this special case show that the algorithm converges to a point where the surrogate models fit the simulator, and they both share the optimum.

Keywords: Derivative-Free Optimization, Non-linear Optimization, Non-convex Optimization, Mixed Integer Programming, Process Simulators, Interpolation Algorithm, Numerical Algorithm.

1. INTRODUCTION

Modern multi-phase flow simulators have been used in both online and off-line applications in the oil industry the last two decades [1, 2]. They have been under continuous development and are today able to solve large flow networks with high accuracy and speed. These attributes motivates a coupling with optimization for production advice and planning. However, the complex nature of such simulators is unfortunate from an optimization point-of-view and complicates implementation. A considerable research effort within the area of non-convex and large-scale optimization during the last years brings new belief in robust implementations of simulator-based optimization. This has motivated the authors to look at how a state of the art optimization solver handles the complexity of a commercial multi-phase flow simulator. Especially, we want to look at a derivative-free approach that utilizes surrogate models and the known structure of the flow network. The high complexity of simulators is often abstracted by regarding it as *one* black-box model, i.e. one input-output map without available gradients [7]. We want to divide it into several smaller black-box models and explicitly express the network structure in the optimization problem.

In an oil and gas production system, where wells are connected to manifolds with multiple pipelines, a complex optimization problem arises. The objective is to maximize the oil production while respecting constraints on the system, such as limited capacity of processing facilities. The production is dependent on advanced flow dynamics and closing/routing of the wells, which result in nonlinearities and integer variables respectively. The problem thus belongs to the class of Mixed Integer Nonlinear Programming (MINLP) problems. Furthermore, the problem is generally non-convex where global convergence is a major challenge. For oil and gas production a considerable amount of uncertainty is associated with reservoir and well models. It is therefore desirable to find a solution close to the operation/starting point to ensure model accuracy.

The preferred MINLP solver for this work was *Bonmin* from the COIN-OR project of IBM [3]. It has been tested on large scale MINLP problems with good results. It has also several features that makes it easier to solve and analyse non-convex problems, such as different integer handling methods.

⋆ The work presented in this paper was supported by FMC Technologies who supplied FlowManager$^{\text{TM}}$ licences and technical assistance.

The simulator of choice was FlowManager™ from FMC Technologies. It is an advanced, multi-purpose, multi-phase flow simulator for off-line and on-line application [8]. It is widely used by the oil industry to monitor subsea oil and gas production. [1]

2. PROBLEM DESCRIPTION

The well cluster in Fig. 1 is designed to test the algorithm presented in this paper. The cluster is modeled in FlowManager™ with model parameters comparable to a real oil field. A simple Black-oil model is used for the fluid. To simplify analysis and verification the cluster is made symmetric; with two identical pipelines and two pairs of identical wells.

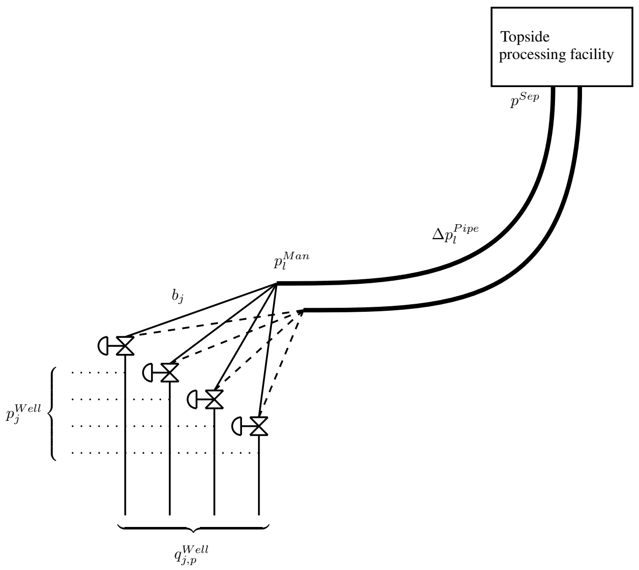

Fig. 1. Cluster with four wells, one manifold, and two pipelines connected to a topside processing facility (first-stage separator). The wells can be routed to either pipeline.

2.1 Simulator model

A commercial multi-phase flow simulator model for the problem in Fig. 1 engulfs complex, non-linear relations between phases and their fluid characteristics, flow rates, pressures, and temperatures. The flow path from downhole to separator is typically split into segments, connected at their boundaries, in which the above relations are calculated and propagated through.

In this paper we consider steady-state models for the wells and pipelines, neglecting near-well reservoir and pipeline flow transients, respectively. We model the wells by assuming that the well performance [2] is given by a static function

$$\bar{p}_j^{Well} = \bar{f}_j(q_{j,oil}^{Well}, q_{j,gas}^{Well}, q_{j,water}^{Well}), \tag{1}$$

[1] At the time of writing, FlowManager™ monitors oil and gas production equivalent to 60% of the production on the Norwegian continental shelf.

[2] In this context the well performance function is a map that gives the *Well Performance Curve* (WPC) of a well.

where f_j gives the relation between the flow rates and wellhead pressure in well j. With a constant Water Cut (WC) and Gas-Oil Ratio (GOR), the above relation can be simplified to

$$\bar{p}_j^{Well} = \bar{f}_j(q_{j,oil}^{Well}), \tag{2}$$

where we only have oil rate as input. This assumption will be used for all wells, with two wells having a GOR equal to 100, two with a GOR of 200, and a common WC of 50%. The well performance curves are given by Figs. 3 and 4.

In a similar manner we describe the pipelines with a static map between flow rates and pressure

$$\Delta\bar{p}_l^{Pipe} = \bar{g}_l(q_{l,oil}^{Pipe}, q_{l,gas}^{Pipe}, q_{l,water}^{Pipe}), \tag{3}$$

where $\Delta\bar{p}_l^{Pipe}$ is the pressure drop in pipeline l, from manifold to separator.

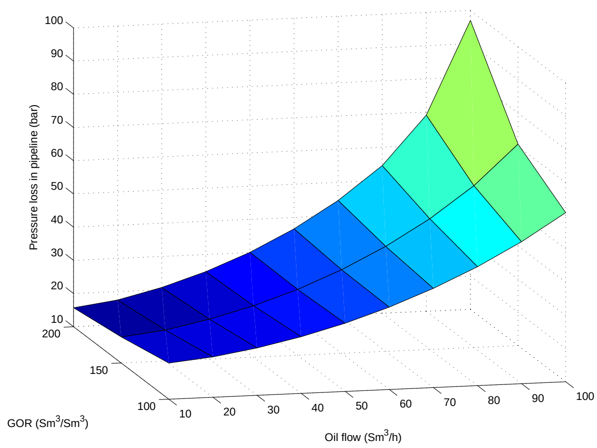

Fig. 2. Pipeline pressure loss for different GORs.

Wells with different GORs and WCs can be routed to the same pipeline. Thus, we cannot reduce the number of inputs to (3) as we did with the well pressure map in (1), resulting in (2). However, a special case appears when all wells have equal WC; the WC of all pipelines will then match the WC of the wells and we can write:

$$\Delta\bar{p}_l^{Pipe} = \bar{g}_l(q_{l,oil}^{Pipe}, q_{l,gas}^{Pipe}). \tag{4}$$

Since all the simulated wells have a GOR between 100 and 200, the interval of possible GORs for the pipelines is $[100, 200]$. Table 1 gives the GOR and WC for all wells and pipelines. Fig. 2 shows the pressure loss in the pipelines for different flow rates.

Table 1. GOR and WC for wells and pipelines

	Wells 1 & 2	Wells 3 & 4	Pipelines
GOR [-]	100	200	[100, 200]
WC [%]	50	50	50

Note that (1) and (3) are independent of temperature and absolute pressure. For (1), these simplifications imply a constant temperature profile through the well and a constant reservoir pressure. While for (3), a constant

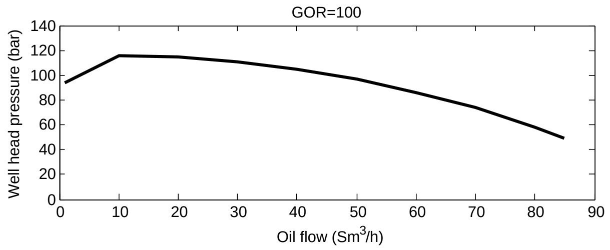

Fig. 3. Well performance curve for wells 1 and 2 (GOR of 100).

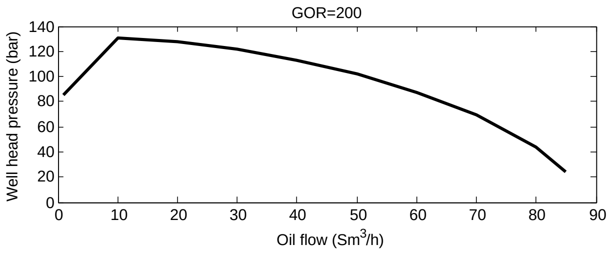

Fig. 4. Well performance curve for wells 3 and 4 (GOR of 200).

temperature profile and a constant separator pressure must be assumed.

The nomenclature used in this paper is given by tables 2, 3, and 4. We will continue to denote the simulator models and their variables with the bar in superscript, to distinguish them from the surrogate models and optimization variables.

Table 2. Index sets

Index and Set	Description
$j \in \mathcal{J} = \{1, 2, 3, 4\}$	Wells
$l \in \mathcal{L} = \{1, 2\}$	Pipelines
$p \in \mathcal{P} = \{\text{oil}, \text{gas}, \text{water}\}$	Phases

Table 3. Variables

Variable	Description	Unit
$b_{j,l}$	Binary routing variable for flow from well j to pipeline l	[-]
$q_{j,p}^{Well}$	Flow rate of phase p from well j	$[m^3/h]$
$q_{l,p}^{Pipe}$	Flow rate of phase p in pipeline l	$[m^3/h]$
p_j^{Well}	Wellhead pressure in well j	[bar]
p_l^{Man}	Pressure in pipeline l at manifold	[bar]
Δp_l^{Pipe}	Pressure drop over pipeline l	[bar]

Table 4. Parameters

Parameters	Description	Unit
p^{Sep}	Pressure at first-stage separator	[bar]
C_{gas}^{Tot}	Gas capacity of first-stage separator	$[m^3/h]$
$r_{GOR,j}$	Gas-Oil Ratio (GOR) for well j	[-]
$r_{WC,j}$	Water Cut (WC) for well j	[%]

2.2 Surrogate models

Assuming a constant GOR and WC, we approximate the wellhead pressure in (2) by a second degree polynomial:

$$\begin{aligned} p_j^{Well} &= f_j(q_{j,oil}^{Well}) \\ &= \alpha_{0,j} + \alpha_{1,j} \cdot q_{j,oil}^{Well} + \alpha_{2,j} \cdot (q_{j,oil}^{Well})^2, \end{aligned} \tag{5}$$

where $\alpha_{0,j}$ is the zero degree coefficient of well j, and so on. In order to make the approximation good we want $e_j^{Well} = p_j^{Well} - \bar{p}_j^{Well} = f_j(q_j^{Well}) - \bar{f}_j(q_j^{Well})$ to be small.

The pipeline pressure drop in (4) is also approximated using a second degree polynomial, that is

$$\begin{aligned} \Delta p_l^{Pipe} &= g_l(q_{l,oil}^{Pipe}, q_{l,gas}^{Pipe}) \\ &= \beta_{0,l} + \beta_{1,l} \cdot q_{l,oil}^{Pipe} + \beta_{2,l} \cdot q_{l,gas}^{Pipe} \\ &\quad + \beta_{3,l} \cdot \left(q_{l,oil}^{Pipe}\right)^2 + \beta_{4,l} \cdot \left(q_{l,gas}^{Pipe}\right)^2 \\ &\quad + \beta_{5,l} \cdot q_{l,oil}^{Pipe} \cdot q_{l,gas}^{Pipe}, \end{aligned} \tag{6}$$

where we have assumed that the WC is equal for all wells. This enables us to express the pressure drop over the pipeline as a function dependent of only oil and gas rate for a given WC. Again, we want the approximation error to be small, i.e. $e_l^{Pipe} = \Delta p_l^{Pipe} - \Delta \bar{p}_l^{Pipe} = g_l(q_l^{Pipe}) - \bar{g}_l(q_l^{Pipe})$.

2.3 Optimization problem

We adopt the following statement of the production optimization problem from [4, 6].

The objective of the optimization is to maximize oil production, that is

$$\underset{\mathbf{x}}{\text{minimize}} \; J(\mathbf{x}) = -\sum_{l \in \mathcal{L}} q_{l,oil}^{Pipe}, \tag{7}$$

where the vector $\mathbf{x}$ represents the decision variables. I.e. $\mathbf{x} = \left[q_{j,p}^{Well}, q_{l,p}^{Pipe}, b_{j,l}, p_j^{Well}, p_l^{Man}\right]^{\mathsf{T}}$, with the variables stacked according to their indices.

The above objective is subject to several constraints that can be categorized into: physical limitations (production capacity), structural constraints (e.g. mass balances), and surrogate model constraints. The structural constraints can be thought of as the links between the surrogate models.

Production capacity constraints. The total production is constrained by the topside processing equipment's (gas and water handling) capacity. In example, we can express the gas handling capacity constraint as

$$\sum_{l \in \mathcal{L}} q_{l,gas}^{Pipe} \leq C^{Tot}, \tag{8}$$

where C^{Tot} is the maximum gas flow rate the processing equipment can handle. In this paper we are interested in pressure constrained solutions [3], where the gas capacity constraint is inactive.

Routing, mass balance, and pressure constraints. A well can be routed to zero or one pipelines (consider it closed when routed to zero pipelines). This routing condition is expressed as

[3] In many subsea production optimization problems the solution is bounded either by pressure or processing capacity constraints.

$$\sum_{l\in\mathcal{L}} b_{j,l} \leq 1, \text{ for } j \in \mathcal{J}. \tag{9}$$

For all phases the mass balance between a well and pipeline can be expressed by

$$q_{l,p}^{Pipe} = \sum_{j\in\mathcal{J}} b_{j,l} \cdot q_{j,p}^{Well}, \text{ for } l \in \mathcal{L}, p \in \mathcal{P}. \tag{10}$$

To ensure a positive pressure drop over the production choke of each producing well we include

$$b_{j,l} \cdot p_l^{Man} \leq p_j^{Well}, \text{ for } j \in \mathcal{J}, l \in \mathcal{L}. \tag{11}$$

Model constraints. For the wells we assume a non-negative flow rate and a constant GOR and WC, i.e.

$$q_{j,oil}^{Well} \geq 0, \tag{12}$$
$$q_{j,gas}^{Well} = r_{GOR,j} \cdot q_{j,oil}^{Well}, \tag{13}$$
$$q_{j,water}^{Well} = \frac{r_{WC,j}}{1 - r_{WC,j}} \cdot q_{j,oil}^{Well}, \tag{14}$$

which holds for all $j \in \mathcal{J}$. The relation between flow rate and pressure in a well is brought into the optimization problem as an equality constraint:

$$p_j^{Well} = f_j(q_{j,oil}^{Well}), \text{ for } j \in \mathcal{J}, \tag{15}$$

where we have assumed a constant GOR and WC, and $f_j(\cdot)$ is given by (5). Similar for the pipelines, we have that:

$$p_l^{Man} - p^{Sep} = g_l(q_{l,oil}^{Pipe}, q_{l,gas}^{Pipe}, q_{l,water}^{Pipe}), \text{ for } l \in \mathcal{L}, \tag{16}$$

where the pressure drop is written as $\Delta p_l^{Pipe} = p_l^{Man} - p^{Sep}$ to avoid unnecessary decision variables in the formulation. The right hand side is given by (6).

Summary. The complete optimization problem, when posed in terms of surrogate models, becomes: Minimize the objective function in (7), subject to the constraints (8) - (16). The problem has 32 decision variables and 33 constraints.

2.4 Data fitting

The surrogate models for the wells are second degree polynomials with three coefficients; $\alpha_{j,0}$, $\alpha_{j,1}$, and $\alpha_{j,2}$. With three data points from the simulator the parameters are selected (uniquely) to make the polynomial go through all points, giving a perfect match.

For the pipelines, cross-terms are assumed to be zero (i.e. $\beta_{l,5} = 0$), simplifying the polynomials to paraboloids with five coefficients; $\beta_{l,0}$ to $\beta_{l,4}$. Each paraboloid is then fitted to the simulator using five data points.

3. ALGORITHM

The algorithm presented in this paper is inspired by a recent work on simulation-based optimization, presented in [5]. It is a derivative-free approach, that is, the algorithm does not have access to any derivatives of the simulator model. The simulators are treated as black-box models and the algorithm can only query the simulators for data points. In this case it can ask for wellhead pressure or pipeline pressure drop by supplying well or pipeline flow rates, respectively.

The algorithm gathers data points to generate local approximations (surrogate models) of the black-box models. A MINLP optimization problem is then established by combining these models with known structural constraints such as mass and pressure balances. The MINLP is solved by *Bonmin*, which handles the integer variables directly in the optimization using Branch&Bound. The nonlinear part of the problem is solved by *Ipopt*, an interior point method within the COIN-OR framework. Ipopt is an integral part of Bonmin.

At the optimum the algorithm compares the surrogate models with the simulator. If they are close, the optimum is also an optimum for the simulator. If not, the surrogate models are updated, and the procedure is rerun. The algorithm is illustrated by Fig. 5 and described by pseudocode in Algorithm 1.

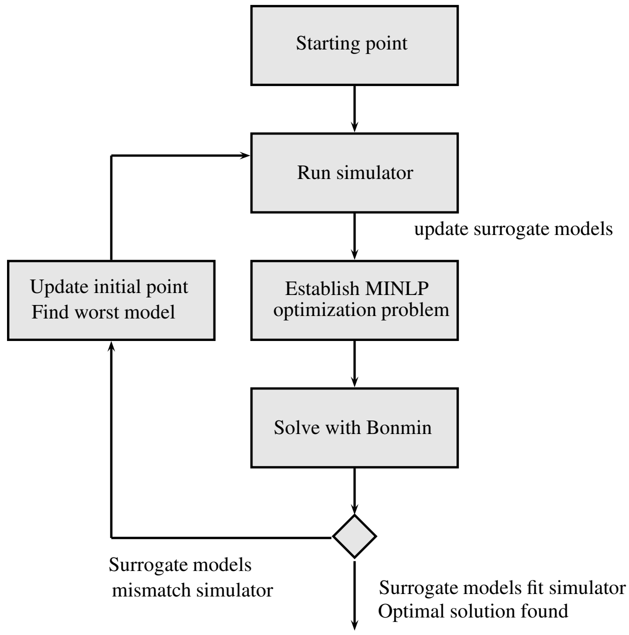

Fig. 5. Algorithmic approach.

Comments on Algorithm 1. Notice the implicit understanding that when the surrogate model is updated at step 8, a new optimization problem arises. In the next iteration, the updated problem is solved by Bonmin; written as the function call $\mathbf{x}_{opt} \leftarrow \text{Bonmin}(\mathbf{x}_0)$.

The algorithm works sequentially by updating *one* surrogate model at each iteration, making it less suitable to parallelization. A computational speed-up may be introduced by updating several models at each iteration, required that the computational cost of the optimization dominates that of the simulators.

Please note that the pseudocode does not include implementation specific details such as initialization handling

Algorithm 1. Algorithm steps

$\mathbf{x}_0 \leftarrow \mathbf{0}$ ▷ Set starting point for Bonmin
Fit surrogate model to simulator around $\mathbf{x}_0$
repeat
  $\mathbf{x}_{opt} \leftarrow \text{Bonmin}(\mathbf{x}_0)$ ▷ Solve opt. problem
  $e^{Well} \leftarrow \max\left\{|f_j(\mathbf{x}_{opt}) - \bar{f}_j(\mathbf{x}_{opt})|^2 \mid j \in \mathcal{J}\right\}$
  $e^{Pipe} \leftarrow \max\left\{|g_l(\mathbf{x}_{opt}) - \bar{g}_l(\mathbf{x}_{opt})|^2 \mid l \in \mathcal{L}\right\}$
  $e \leftarrow \max\left(e^{Well}, e^{Pipe}\right)$ ▷ Largest error
  Fit surrogate model associated with e to simulator data points around $\mathbf{x}_{opt}$
  $\mathbf{x}_0 \leftarrow \mathbf{x}_{opt}$ ▷ Next starting point for Bonmin
until $e \leq \epsilon$ ▷ End loop when largest error is small
return $\mathbf{x}_{opt}$ ▷ Optimal point found

and feasibility checks. Furthermore, the authors have taken the liberty to simplify the pseudocode by writing $f(\mathbf{x}_{opt})$ and $g(\mathbf{x}_{opt})$, although f and g are functions of only some flow rates in $\mathbf{x}$.

4. RESULTS

Applying Algorithm 1 on the test case gave us the following results. The convergence of the algorithm is presented in Fig. 6. The algorithm converges rapidly towards the point of "global" optimal production [4] . After six iterations the optimum is found, and after only three iterations the solution is close to the optimum. This was achieved even with an infeasible starting point with all decision variables equal to zero, due to the infeasibility check implemented in Bonmin (consult [3] for details).

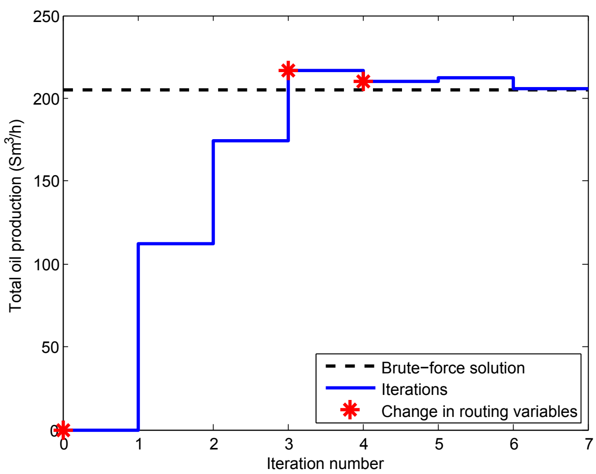

Fig. 6. Convergence of algorithm when $\mathbf{x}_0 = \mathbf{0}$.

Bonmin changes routing variables three times without deteriorating the solution (internally, Bonmin tests many more routing combinations for each MINLP problem it solves). This behaviour is due to the updating of the surrogate models and the fact that Bonmin solves different MINLP problems at each iteration. Also, remark that some of the solutions are above the optimal production we compare with. This is simply a result of mismatch between the surrogate models and simulators.

At each iteration a surrogate model is updated. In this case the pipeline models had the largest error in all iterations and were thus the only models that were updated. This proves that the polynomial surrogate models are good approximations of the well performance curves. As shown in Figs. 7 and 8, the approximations are good around the optimal point.

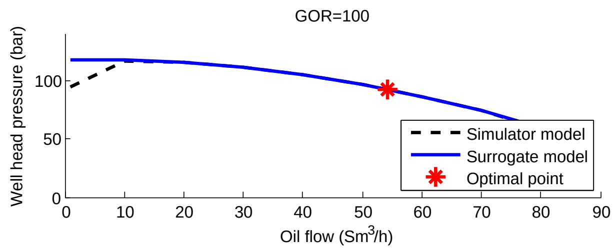

Fig. 7. Final surrogate model and optimal point for wells 1 and 2 with a GOR of 100.

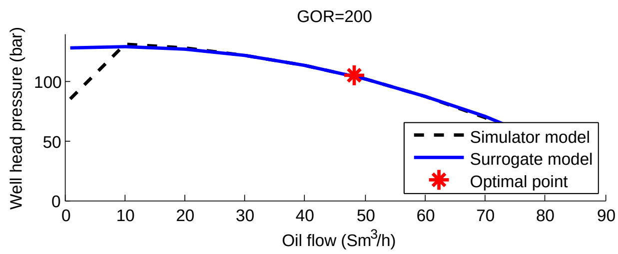

Fig. 8. Final surrogate model and optimal point for wells 3 and 4 with a GOR of 200.

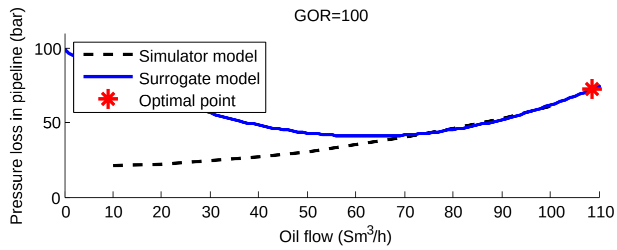

Fig. 9. Final surrogate model and optimal point for pipeline 1. GOR is 100.

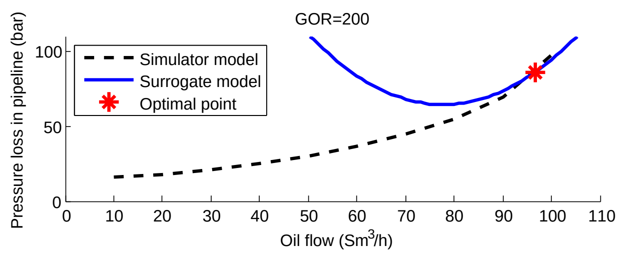

Fig. 10. Final surrogate model and optimal point for pipeline 2. GOR is 200.

[4] The optimal production was found by running a brute-force Sequential Quadratic Programming (SQP) algorithm that, given a starting point, solved the NLP problems resulting from locking the routing variables at every possible combination. The best solution found is regarded as globally optimal in the sense of routing variables. The solutions were validated in FlowManager^{TM}.

Figs. 9 and 10 show the final surrogate models of the pipelines together with the optimum. Wells with equal GOR are routed to the same pipeline at the optimum, and we thus plot pipeline 1 for a GOR of 100 and pipeline 2 for a GOR of 200. Unlike the surrogate models of the wells, the pipeline surrogate models only approximate the simulator locally, as seen from the high curvature.

5. CONCLUSION

The results from the previous chapter displays that for this specific case, Algorithm 1 converges to an optimal point, without using gradient information on the wells and pipelines. The algorithm was able to reject many of the integer combinations and converged with few iterations.

The algorithm needed 7 iterations to converge, and considered three different routing alternatives. However, the nature of a MINLP solver is to implicitly consider all routing alternatives by relaxing the integer variables. Comparing it to a brute force approach regarding number of routing alternatives tested, is therefore not completely fair. The proposed algorithm solves a MINLP problem in each iterations, while the brute force approach solves a NLP problem for each routing combination. However, as the number of wells increases, so does the number of NLP problems that needs to be solved in the brute force approach, and testing all alternatives fast becomes intractable. For the algorithm proposed in this paper, we do not expect the number of iterations to increase in the same way, due to the implicitly handling of the routing variables. And hence, the advantage should increase with the complexity of the subsea production networks.

The core of the algorithm is the MINLP solver Bonmin. The solver showed great robustness to solve non-convex problems of class MINLP using Branch&Bound. The surrogate models gave good local approximations of the simulators, especially in the friction dominated (high rate) flow region of the wells.

The authors encourage further development of the algorithm in the directions outlined below.

5.1 Further research

It is of great interest to improve the accuracy and robustness of the algorithm for more involved applications. The authors identify the following areas of interest for further research.

Surrogate models. In this paper all surrogate models are second degree polynomials. It would be interesting to investigate other model classes. The choice of surrogate model class is entirely dependent on the simulator model that is to be approximated.

Also, for simulators with discontinuities due to internal model switching (e.g. changing flow regime i multi-phase flow simulators) it could be of interest to switch surrogate model. This can be done by including binary switching variables to the problem.

Model fitting. Two questions that naturally occur when performing model fitting are: one, how many data points to use, and two, how to pick those points.

In this work we selected the minimum number of points and fit the surrogate models exactly to these. If the simulator model is prone to some sort of noise, discontinuities, or severe non-linearities, the bare minimum of points may result in a fit that is bad, even locally. It would then be better to perform a weighted least-square fit on more points.

The second question leads to a trade-off situation where the distribution of points balances between a locally or globally good fit. For the algorithm treated in this paper it could be a good approach to start out with a global approach and move towards a local approach as the algorithm seeks to the optimum.

Trust region. Convergence and feasibility problems can occur when the model is updated at each time step. The authors believe that by incorporating a trust region to the surrogate models, keeping them close to the simulator model and within the feasibility region, one can avoid some of these problems.

ACKNOWLEDGEMENTS

We acknowledge the support of the Center for Integrated Operations at NTNU, Norway.

REFERENCES

[1] Bieker, H.P., Slupphaug, O., and Johansen, T.A. (2006). Real-Time Production Optimization of Offshore Oil and Gas Production Systems: A Technology Survey. *Proceedings of Intelligent Energy Conference and Exhibition.*

[2] Bringedal, B.r., Storkaas, E., Dalsmo, M., Aarset, M., With, H.M., and Group, S.P.T. (2010). Recent Developments in Control and Monitoring of Remote Subsea Fields. *Proceedings of Intelligent Energy Conference and Exhibition.*

[3] COIN-OR (2011). *COmputational INfrastructure for Operations Research, http://www.coin-or.org/, 24.10.2011.*

[4] Foss, B., Gunnerud, V., Diez, M.D., and Asa, S. (2009). Lagrangian Decomposition of Oil- Production Optimization Applied to the Troll West Oil Rim. *SPE Journal.*

[5] Gunnerud, V., Conn, A., and Foss, B. (2012). Embedding structural information in simulation-based optimization. *Pending paper.*

[6] Gunnerud, V. and Foss, B. (2010). Oil production optimization - A piecewise linear model, solved with two decomposition strategies. *Computers & Chemical Engineering.*

[7] Kramer, O., Ciaurri, D., and Koziel, S. (2011). Derivative-free optimization. *Computational Optimization, Methods and Algorithms*, 61–83.

[8] Lunde, G.G., Vannes, K., Mcclimans, O.T., Technologies, F.M.C., Burns, C., and Wittmeyer, K. (2009). Advanced Flow Assurance System for the Ormen Lange Subsea Gas Development. *Offshore Technology Conference.*