Global optimization of multiphase flow networks using spline surrogate models

Bjarne Grimstad[a,*], Bjarne Foss[a], Richard Heddle[b], Malcolm Woodman[b]

[a]*Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway.*

[b]*BP Exploration Operating Company Limited, Chertsey Road, Sunbury on Thames, Middlesex, TW16 7BP.*

Abstract

A general modelling framework for optimization of multiphase flow networks with discrete decision variables is presented. The framework is expressed with the graph, and special attention is given to the convexity properties of the resulting programming formulation. Nonlinear pressure and temperature relations are modelled using multivariate splines and a special mixed-integer nonlinear programming (MINLP) formulation with spline constraints results. A global solution method is devised by combining the framework with a spline-compatible MINLP solver, recently presented in the literature. The solver is able to globally solve the nonconvex optimization problem. The new solution method is benchmarked with several local optimization methods on a set of three realistic subsea production optimization cases provided by the oil company BP.

Keywords: nonlinear flow networks, petroleum production optimization, mixed-integer nonlinear programming, branch-and-bound, splines.

1. Introduction

Multiphase flow networks appear in many application areas. In this paper we are particularly interested in multiphase flow networks for subsea oil and gas production. Such networks consist of wells, collection systems, pipelines, and in some cases processing units such as pumps and separators. In recent years real-time data capture and storage capabilities have become an industry standard, thus paving the way for the use of model-based techniques to improve operations. In practice, the use of model-based methods translates into advisory systems for production engineers. Such systems use real-time data in combination with calibrated mathematical models and optimization to improve economics of an oil field by increasing oil throughput. It can be hard to measure the true value of model-based advisory systems since they have impact on profit, cost of operating, HSE and operating risk, and possibly other non-economic values. This may explain why operators tend to prefer maximization of hydrocarbons (oil and gas): hydrocarbon production can be measured, and sometimes must be measured to conform with legislation and fiscal systems. Some claims to a production increase in the range from 1-4% due to use of model-based tools can be found in the literature (Stenhouse et al., 2010; Teixeira et al., 2013). In the latter

[*] Corresponding author.
Email address: `bjarne.grimstad@gmail.com` (Bjarne Grimstad)

testimonial, a 1.2% production increase on a medium size offshore production vessel is claimed, amounting to \$35mill per year. Thus, the economic potential is clearly significant. Despite this, real-time decision support tools as alluded to above are rarely used in the upstream petroleum industries.

Two key reasons for limited use are lack of tools for model maintenance, and robust and efficient solvers, respectively. First, models must be updated periodically due to the time-varying nature of the production system; in particular, reservoir conditions change with time due to reservoir pressure decline and changing fluid compositions. Second, the optimization problem itself is hard to solve since models are nonlinear and often available only as black box calculators. In fact, oil and gas production systems are typically modelled in proprietary process simulators, not offering gradient information. Thus, there are several factors that contribute to long solution times, including the following: a lack of analytical derivatives, computational expensive evaluations of the process simulator, and slow IO operations in the communication between process simulator and optimization solver. It may be added that different parts of the flow network, in particular well models and pipeline models, may be available in different simulator applications, thus complicating matters even more. Moreover, decision variables are both continuous and discrete. Thus, we are faced with mixed-integer nonlinear (MINLP) problems that may include black-box constraints. Long solution times prevent efficient use of decision support tools and break the natural workflow of the production engineers. When it takes several hours to arrive at an optimization the result is often "out of date" before it is available to the them.

This paper suggests a methodology to overcome the challenges related to the optimization part as presented above. This is done in three steps. First, we adapt a well known, graph-based modelling scheme to oil and gas networks. Second, we propose the use of spline-based surrogate models to represent the nonlinear parts of the system. This implies that models, which are available as proprietary (black-box) simulators, explicit model equations or look-up tables, are approximated with splines through a sampling and interpolation scheme. By performing this substitution for each item of equipment in the network *a priori* optimization, the solver can be decoupled from the process simulator during the optimization run, resulting in a considerable reduction in solution times. Third, we introduce a global branch-and-bound based MINLP solver that exploits the facts that all nonlinearities are described by splines and takes advantage of the structural properties of oil and gas networks. In order to evaluate our approach it was deemed necessary to use a comprehensive and realistic test bench rather than simplistic cases. Thus, three industrial cases are used where all relevant models and constraints are included.

The remainder of this paper is organized as follows. A short description of the production optimization problem for subsea production systems is given in Section 2. A brief report on recent works on this topic follows in Section 3. A new mathematical programming framework for optimizing general flow networks is derived in detail in Section 4. In Section 5, we give a description of B-splines, which are used as surrogate models for the nonlinear functions in the problem formulation presented in Section 4. The solution method is presented in Section 6 and benchmarked on several realistic cases in Section 7. Finally, some concluding remarks are given in Section 8.

2. Problem description

Consider the subsea production system illustrated in Figure 1, consisting of reservoirs, wells, manifolds, flowlines, risers, and separators. The system is built to allow a safe and efficient transportation of reservoir fluid to the surface. At the surface the fluid is separated before it is further treated in processing facilities. The fluid flow through the system is controlled with valves, e.g. chokes (adjustable valves) and manifold valves (on/off valves). The valve settings decide the production and operational status of the wells: that is, the flow rate of each well, the routing of the flows through the network, and the allocation of lift gas.

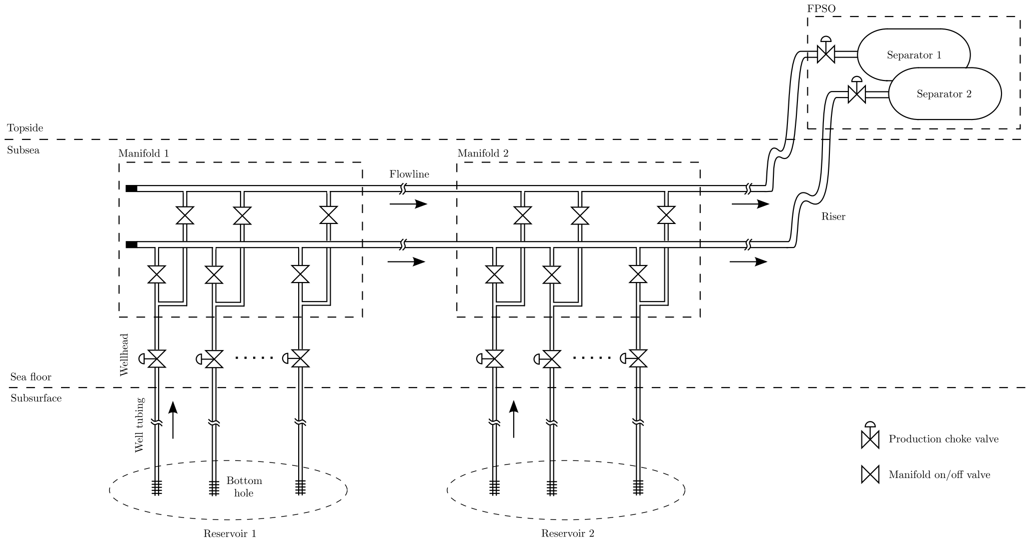

Figure 1: A subsea production system with two daisy-chained manifolds.

The *daily production optimization problem* is the search for valve settings that maximize the production of oil (or profit) while respecting physical laws and operational constraints. Physical laws that must be abided include: mass, momentum, and energy conservation laws; and well inflow relations. Some physical laws may be empirically modelled because of their high complexity. For example, pipeline pressure drops are often modelled with the empirical Beggs and Brill correlation (Beggs et al., 1973). Typical operational constraints may include: upper and lower rate constraints; draw-down (minimum) pressure constraints; oil, gas, and water handling capacity constraints; upper bound on gas-lift availability; and number of allowed routing changes. The operational constraints are typically provided by the user to be in accordance with the current production plan.

A requirement for solving the daily production optimization problem, and for it to provide applicable and optimal solutions, is that an accurate model of the production system is available. The model should accurately predict flow rates for any valve setting in the search space of interest. To reduce the modelling effort, we present a mathematical programming framework that includes the above-mentioned physical laws and operational constraints of a generic production system. The flexible framework, presented in the next section, allows for modelling of most common subsea production system topologies.

3. Previous work

Network flow and design problems lie in the intersection of several domains, including: operations research, applied mathematics, engineering, and computer science (Ahuja et al., 1993).

A diverse set of problems can be formulated as network problems, for example: optimization of urban public transportation networks (Mandl, 1980), train routing and scheduling (Cordeau et al., 1998), and design of optimal water distribution systems (Alperovits and Shamir, 1977).

The recent works of Luathep et al. (2011) and Raghunathan (2013) look at global optimization of the network design problem using a MILP and MINLP approach, respectively. These methods have features similar to the method presented later in this paper. For instance, the graph is used as a modelling tool, convex formulations are obtained by using the big-M relaxation, nonlinear relations are approximated with piecewise functions, and a MILP or MINLP problem is solved using a specialized solver.

In the following we provide a brief report on works that address the petroleum production optimization problem described in Section 2. To the authors' knowledge, the works of Kosmidis et al. (Kosmidis et al., 2004, 2005) were the first to address well oil rate allocation, gas lift allocation, and well routing with a single problem formulation. In these works the complete production optimization problem is posed as a MINLP problem.[1] Many earlier works have considered optimization of individual network components, for instance optimal gas lift allocation on a well basis (Wang, 2003; Rashid et al., 2012). A survey on these early works is provided by Kosmidis et al. (2005) and Bieker et al. (2007). From 2006 and onwards, several works have emerged that build on the contributions of Kosmidis et al. (2005) or use a similar approach, see for example Martin et al. (2006); Misener et al. (2009); Gunnerud and Foss (2010); Codas and Camponogara (2012). These works use piecewise linear formulations to approximate nonlinear relations in the network such as nonlinear pressure drop functions. This modelling approach results in a MILP that scales poorly when nonseparable functions of four or more variables are approximated (Misener and Floudas, 2010; Vielma et al., 2010; Vielma and Nemhauser, 2011). This may be one reason for the relatively simple formulations used in these works. For example, temperatures are not considered and pressure drops are modelled as functions of flow rates only. Interestingly, the MINLP formulation of Kosmidis et al. (2005) did include linear temperature drop models. A computational analysis of different multidimensional piecewise linear models was recently provided by Silva and Camponogara (2014). The analysis shows that SOS2 models and MILP models with a logarithmic number of binary variables have the best performance. These formulations may allow modelling of (nonseparable) functions of four or five variables, for which separation to multiple lower-dimensional functions is not possible.

The spline-based approach presented in this paper can be viewed as an alternative to the piecewise linear approaches mentioned above. The approach to be presented results in a global NLP problem (MINLP if routing is included). The main difference is that the solver must branch on continuous variables, instead of SOS2 or binary variables as is the case in the MILP approaches. The approach allows us to accurately approximate nonlinear functions in up to five variables. This enables us to model pressure and temperature drops as functions of flow rates, pressure, and temperature.

4. Multiphase flow network modelling

In this section we present a general mathematical programming formulation for multiphase flow networks. The goal is to achieve a formulation that is as simple as possible, while capturing

[1] The problem is also referred to as the *daily well scheduling problem* in the literature.

important physics with sufficient accuracy. This goal reflects our desire to obtain a problem formulation that we can solve in reasonable time to get applicable solutions.

The problem formulation is based on the following assumptions:

A1. The system operates at steady-state conditions.

A2. Continuous and differentiable multiphase pressure drop and temperature drop models.

A3. The thermodynamics can be modelled under the assumptions in Section 4.3.

A4. No uncertainty is considered in the model structure or its parameters.

An argument for assumption A1 follows: The daily production optimization problem has a horizon spanning several hours to one day. In general, the fluid dynamics in the network (wells and pipelines) have time constants in the order of minutes, and will appear instant on this horizon. Similarly, the dynamics at the system boundaries appear constant on this horizon: the reservoir dynamics have time constants in the order of weeks to months, and the surface facility dynamics have time constants in the order of seconds to minutes. One may note that there are exceptions where A1 does not hold, examples include dynamic phenomena like slugging or casing-heading instability which are highly influential on the production and act in the relevant horizon of hours.

Assumption A2 ensures continuous and differentiable constraint functions, which is a prerequisite for most gradient-based optimization solvers. As will be discussed, spline surrogate models have these properties by construction, even when the function they approximate do not. Assumption A4 is included since uncertainty is not structurally treated in the proposed framework.

A directed graph $G = (\mathbf{N}, \mathbf{E})$, with nodes $\mathbf{N}$ and edges $\mathbf{E}$, is used to represent the flow network (Ahuja et al., 1993). A node in $\mathbf{N}$ represents a junction or simply a point of interest in the network. An edge in $\mathbf{E}$ connects two nodes and represents a pipe segment (e.g. a wellbore, jumper, flowline, or riser), a valve (e.g. a production choke or manifold valve), or any item of equipment (e.g. a subsea multiphase pump). Valves represent special edges since they can be closed to disjoint the neighbouring nodes. To make this distinction clear we introduce a subset of edges, $\mathbf{E}^{\mathrm{d}}$, that represent the valves. An edge in $\mathbf{E}^{\mathrm{d}}$ is referred to as a *discrete* edge since it has two states: it is either open or closed. Associated with each discrete edge is a binary variable which is used to model the switching between the open and closed state. The discrete edges are used to route the flow through the network by restricting flow through certain valves. All other edges ($\mathbf{E} \backslash \mathbf{E}^{\mathrm{d}}$) represent pipes or equipment.

Table 1 gives the various sets used to describe the flow network. Some utility sets that simplify the notation are given in Table 2. In the rest of this paper the terms *graph*, *network*, and *system* are used interchangeably.

The following requirements are placed on the graph structure:

R1. A source node $i \in \mathbf{N}^{\mathrm{src}}$ has *zero* entering edges and *one* leaving edge, i.e. $\mathbf{E}_i^{\mathrm{in}} = \emptyset$ and $|\mathbf{E}_i^{\mathrm{out}}| = 1$.

R2. A sink node $i \in \mathbf{N}^{\mathrm{snk}}$ has *zero* leaving edges, i.e. $\mathbf{E}_i^{\mathrm{out}} = \emptyset$.

R3. An internal node $i \in \mathbf{N}^{\mathrm{int}}$ has *one or more* leaving edges. It may have *more than one* leaving edges iff all of them are discrete edges and at most one of them can be open at any time.

The first and second requirement follow the normal definition of source and sink nodes. The additional requirement that a source node may have only one leaving edge is made without loss of generality (an equivalent graph fulfilling this requirement can always be obtained by adding nodes). The third requirement on the internal nodes is needed because splitting of fluids

Table 1: Sets

Set	Description
$\mathbf{N}$	Set of nodes in the network.
$\mathbf{N}^{\text{src}}$	Set of source nodes in the network. $\mathbf{N}^{\text{src}} \subset \mathbf{N}$.
$\mathbf{N}^{\text{snk}}$	Set of sink nodes in the network. $\mathbf{N}^{\text{snk}} \subset \mathbf{N}$.
$\mathbf{N}^{\text{int}}$	Set of internal nodes in the network. $\mathbf{N}^{\text{int}} = \mathbf{N} \backslash \{\mathbf{N}^{\text{src}} \cup \mathbf{N}^{\text{snk}}\}$.
$\mathbf{E}$	Set of edges in the network. An edge $e = (i, j)$ connects node i to node j, where $i, j \in \mathbf{N}$.
$\mathbf{E}^{\text{d}}$	Set of *discrete* edges that can be *open* or *closed*. $\mathbf{E}^{\text{d}} \subset \mathbf{E}$.
$\mathbf{S}$	Set of flow phases in the network. For three-phase petroleum flow the phases are denoted $\mathbf{S} = \{\text{oil}, \text{gas}, \text{wat}\}$, where oil denotes the hydrocarbon liquid phase, gas the hydrocarbon gas phase, and wat the water liquid phase. A compositional model may have more than three phases/components.

is not modelled; this simplifying requirement is commonly applied in works on flow network modelling (Codas et al., 2012). This requirement is enforced by manifold routing constraints, to be presented later in this section.

The sets $\mathbf{E}^{\text{src}}$ and $\mathbf{E}^{\text{snk}}$ in Table 2 are *cut-sets*. A cut-set is a set of edges that, if removed, partitions the graph nodes into two disconnected subsets. These sets are useful because the net flow through the graph can be measured as the net flow over the edges in a cut-set.

Table 2: Utility sets

Set	Description
$\mathbf{E}_i^{\text{in}}$	Set of edges entering node i, i.e. $\mathbf{E}_i^{\text{in}} = \{e : e = (j, i) \in \mathbf{E}\}$.
$\mathbf{E}_i^{\text{out}}$	Set of edges leaving node i, i.e. $\mathbf{E}_i^{\text{out}} = \{e : e = (i, j) \in \mathbf{E}\}$.
$\mathbf{E}^{\text{src}}$	Set of edges leaving a source node in $\mathbf{N}^{\text{src}}$, i.e. $\mathbf{E}^{\text{src}} = \bigcup\limits_{i \in \mathbf{N}^{\text{src}}} \mathbf{E}_i^{\text{out}}$.
$\mathbf{E}^{\text{snk}}$	Set of edges entering sink node in $\mathbf{N}^{\text{snk}}$, i.e. $\mathbf{E}^{\text{snk}} = \bigcup\limits_{i \in \mathbf{N}^{\text{snk}}} \mathbf{E}_i^{\text{in}}$.
$\mathbf{N}^{\text{d}}$	Set of nodes with discrete leaving edges, i.e. $\mathbf{N}^{\text{d}} = \{i : i \in \mathbf{N}, \mathbf{E}_i^{\text{out}} \subset \mathbf{E}^{\text{d}}\} \subset \mathbf{N}$.

The variables of the problem, listed in Table 3, are related to the nodes and edges of the graph. The flow rates are given as mass flow rates or as volumetric flow rates in standard conditions. In the latter case, the flow rates must be properly scaled with the phases' standard condition densities, denoted with ρ_s for $s \in \mathbf{S}$. For brevity, the phase flow rates on an edge $e \in \mathbf{E}$ are collectively denoted $\mathbf{q}_e$, that is, with an oil, gas, and water phase, $\mathbf{q}_e = [q_{e,\text{oil}}, q_{e,\text{gas}}, q_{e,\text{wat}}]^{\mathsf{T}}$. Furthermore, we denote all the flow rates, pressures, and pressure drops in the network with $\mathbf{q}$, $\mathbf{p}$, and $\mathbf{\Delta p}$, respectively. We use the same notation for vectors containing the temperature and enthalpy variables.

With the network topology represented by the graph, and the variables and parameters associated with the nodes and edges, the flow network is modelled by placing control volumes around each node and edge. In each control volume *mass*, *momentum*, and *energy* conservation laws are enforced. In the following, we present the equations/constraints for the conservation laws, as well as some operational constraints. Together with an objective, they form the basis

Table 3: Variables

Variable	Description
p_i	Pressure at node $i \in \mathbf{N}$.
Δp_e	Pressure drop over edge $e = (i, j) \in \mathbf{E}$, e.g. $\Delta p_e = p_i - p_j$.
t_e	Temperature of fluid entering edge $e \in \mathbf{E}$.
Δt_e	Temperature drop over edge $e \in \mathbf{E}$. The fluid leaving e has a temperature of $t_e - \Delta t_e$.
h_e	Enthalpy of fluid entering edge $e \in \mathbf{E}$.
Δh_e	Enthalpy drop over edge $e \in \mathbf{E}$. The fluid leaving e has an enthalpy of $h_e - \Delta h_e$.
$q_{e,s}$	Flow rate of phase $s \in \mathbf{S}$ on edge $e \in \mathbf{E}$.
y_e	Binary variable associated with an edge $e \in \mathbf{E}^{\mathrm{d}}$. If $y_e = 1$ the edge is open, allowing a nonzero flow; otherwise, $y_e = 0$ and the edge is closed with zero flow.

Table 4: Parameters

Parameter	Description
$q^L_{e,s}, q^U_{e,s}$	Lower and upper bound, respectively, for flow rate $q_{e,s}$ of phase $s \in \mathbf{S}$ on edge $e \in \mathbf{E}$. It is assumed that $0 \leq q^L_{e,s} \leq q^U_{e,s}$.
p^L_i, p^U_i	Lower and upper bound, respectively, for pressure p_i in node $i \in \mathbf{N}$. It is assumed that $0 \leq p^L_i \leq p^U_i$.
t^L_e, t^U_e	Lower and upper bound, respectively, for temperature t_i on edge $e \in \mathbf{E}$. It is assumed that $0 \leq t^L_e \leq t^U_e$.

of the proposed mathematical programming problem formulation, or framework, for production optimization. The complete formulation is given towards the end of this section.

4.1. Mass balances

In steady-state, the mass flow into a node must equal the mass flow out of it, i.e., there is no accumulation of fluid in the node (or in the network). Using the sets $\mathbf{E}^{\mathrm{in}}_i$ and $\mathbf{E}^{\mathrm{out}}_i$, the mass balances for the nodes may be expressed as:

$$\sum_{e \in \mathbf{E}^{\mathrm{in}}_i} q_{e,s} - \sum_{e \in \mathbf{E}^{\mathrm{out}}_i} q_{e,s} = 0, \quad \forall s \in \mathbf{S}, i \in \mathbf{N}^{\mathrm{int}}. \tag{1}$$

Note that the mass balances are defined only for the *internal* nodes in the network ($\mathbf{N}^{\mathrm{int}}$). Since a source (sink) node have leaving (entering) edges only, its mass balance would enforce zero net flow out (in) of the node. Hence, source and sink nodes are excluded from Eq. (1).

4.2. Momentum balances

The multiphase flows in the network are driven by the node pressures (potentials) p_i. The pressure drop over an edge $e = (i, j)$ is defined as $\Delta p_e \triangleq p_i - p_j$, and relates the two node pressures p_i and p_j. For edges $e \in \mathbf{E}^{\mathrm{d}}$ that represent choke or on/off valves, Δp_e is a free/adjustable variable as discussed in Appendix A. For edges $e \in \mathbf{E} \backslash \mathbf{E}^{\mathrm{d}}$ that represent pipes, Δp_e is given by some pressure drop correlation $\Delta p_e = f_e(\mathbf{q}_e, p_i, t_e)$. The function $f_e(\cdot)$ maps the upstream conditions (flow rates, pressure, and temperature) to the pressure drop Δp_e. When it is more convenient to express f_e in terms of the downstream pressure, p_i can simply be replaced with p_j. The pressure drops in the network are modelled with the following constraints:

$$\Delta p_e = f_e(\mathbf{q}_e, p_i, t_e), \quad \forall e \in \mathbf{E} \backslash \mathbf{E}^{\mathrm{d}}. \tag{2}$$

Notice that Eq. (2) does not apply to edges with an adjustable pressure drop ($\mathbf{E}^{\mathrm{d}}$). A pressure correlation may be insensitive to temperature for certain flow conditions, e.g. liquid dominated flows. In this case the correlation can be simplified to $\Delta p_e = f_e(\mathbf{q}_e, p_i)$ without any significant loss of accuracy. Another special case occurs for edges representing short pipes with negligible pressure drop, i.e. with $f_e(\cdot) \approx 0$, giving $\Delta p_e = p_i - p_j \approx 0$. In the rest of this paper $f_e(\cdot)$ will be used to denote the pressure correlation of edge e, even if $f_e(\cdot) = 0$.

For a discrete edge $e \in \mathbf{E}^{\mathrm{d}}$, the momentum balance needs to be deactivated when the edge is closed. This logic is accurately expressed by the following disjunction:

$$\begin{bmatrix} y_e = 0 \end{bmatrix} \vee \begin{bmatrix} y_e = 1 \\ \Delta p_e = p_i - p_j \end{bmatrix}. \tag{3}$$

The disjunction in Eq. (3) can be interpreted as follows: if an edge e is closed ($y_e = 0$), then there is no *direct* relation between the pressures in the adjacent nodes i and j (the node pressures may still be *indirectly* related through other paths in the network); if the edge is open ($y_e = 1$), then the two pressures must satisfy the relation $\Delta p_e = p_i - p_j$, where Δp_e is given by $f_e(\cdot)$ in Eq. (2).

Although the disjunction in Eq. (3) captures the desired logic for the momentum balance its form is not widely supported by commercial solvers. A straightforward way to deal with the disjunction without using logical expressions is to approximate it with

$$y_e(p_i - p_j - \Delta p_e) = 0. \tag{4}$$

This formulation introduces an additional (and undesired) nonlinearity to the problem through the multiplication with y_e. This nonlinearity can be relaxed using linear big-M constraints.[2] Notice that the pressures p_i and p_j are constrained to $0 \leq p_i^L \leq p_i \leq p_i^U$ and $0 \leq p_j^L \leq p_j \leq p_j^U$. In practice p_j^L and p_j^U may be inferred from p_i^L and p_i^U, and the image of $f_e(\cdot)$. These bounds imply that $-M_e \leq p_i - p_j - \Delta p_e \leq M_e$, where $M_e = (p_i^U - p_i^L) + (p_j^U - p_j^L)$. Using M_e, the disjunction in Eq. (3) may be approximated with the big-M constraints

$$-M_e(1 - y_e) \leq p_i - p_j - \Delta p_e \leq M_e(1 - y_e). \tag{5}$$

For $y_e = 1$, Eq. (5) yields $0 \leq p_i - p_j - \Delta p_e \leq 0$, and the constraint $\Delta p_e = p_i - p_j$ in Eq. (3) is retrieved. For $y_e = 0$, Eq. (5) yields two constraints which are inactive in the feasible set: e.g., the inactive constraints allow Δp_e to take on any value in $[(p_i^L - p_j^U), (p_i^U - p_j^L)]$, effectively disconnecting p_i and p_j. Thus, the relaxation do not alter the optimal solution of the problem. A drawback with using big-M constraints is that they often produce a weak relaxation of the disjunction.[3] However, reasonably tight values for M_e can easily be derived from the pressure drop functions. Thus, we accept Eq. (5) as an alternative model to Eq. (3) and use it to model the momentum balances. Before proceeding, we note that the same big-M constraints were used by Codas et al. (2012). We also note that an alternative relaxation could have been achieved by using a convex hull formulation (Grossmann, 2002).

[2] The big-M constraints can easily be derived by applying McCormick's relaxation of bilinear terms to Eq. (4).

[3] In theory it is possible to let $M_e \to \infty$ and still obtain a valid relaxation. This will however give an increasingly poor relaxation and produce ill-conditioned systems of equations in the solver, causing numerical problems.

4.3. Energy balances

The thermodynamic potentials (enthalpies), h_e, of the fluids in the network are modelled using the temperature variables t_e. At the source nodes, fluid enters the network with a specified temperature (typically close to the reservoir temperature). As the fluid flows through the network, its temperature changes due to mixing with other fluids and energy loss through the pipe walls to the surroundings. In the following we will assume that the mixing happens at the nodes, and the energy loss occurs at the edges (pipes). To simplify the modelling we make the following assumptions:

- Instant mixing. At any point in the network all fluid phases are assumed to have the same temperature.
- No work is done by the system. However, the model can easily be extended to include energy generation or loss through work, allowing for active components such as pumps and compressors.
- The heat transfer between the system and its surroundings may be completely determined from internal states. Consequently, the heat transfer properties and surrounding temperatures are assumed to be constant.
- The enthalpy is equal to the internal energy of a fluid, that is, no pV-work is done. This assumption is reasonable for a stationary process without fluid accumulation.
- Constant heat capacities c_s for all phases $s \in \mathbf{S}$.

Regarding the last assumption above: In general, the heat capacity of a fluid is a function of pressure and temperature, i.e. $c_s = c_s(p_i, t_e)$. In practice, this relation is available in a compositional PVT model or a black oil model (Aziz and Settari, 1979). To simplify the model we assume c_s to be constant in this work. This simplification is reasonable for liquids, but may give rise to large errors for gases. However, in the enthalpy calculations below, the contribution from gas is generally much smaller than that of liquids, mitigating the erroneous heat capacity of gas. According to the above assumptions, we next present the equations in the thermodynamic model.

The temperature drop over the edges are modelled as

$$\Delta t_e = g_e(\mathbf{q}_e, p_i, t_e), \quad \forall e \in \mathbf{E}. \tag{6}$$

The relation gives the temperature change due to heat transfer through the pipe walls to the surroundings. In short, insulated, non-restrictive pipes the temperature drop can usually be ignored by setting $\Delta t_e = 0$.

The enthalpy of the fluid entering edge e, and the change in enthalpy across edge e, are calculated as

$$\begin{aligned} h_e &= t_e \sum_{s \in \mathbf{S}} c_s \cdot q_{e,s}, &\quad \forall e \in \mathbf{E}, \\ \Delta h_e &= \Delta t_e \sum_{s \in \mathbf{S}} c_s \cdot q_{e,s}, &\quad \forall e \in \mathbf{E}, \end{aligned} \tag{7}$$

where c_s is the constant heat capacity of fluid $s \in \mathbf{S}$.[4] As previously mentioned, nonlinear heat capacities on the form $c_s(p_i, t_e)$ may be used in Eq. (7) to increase the accuracy of the model. However, for liquid dominated flow the contribution to enthalpy from gas is relatively small.

[4] In Eq. (7) the heat capacities c_s are given in [J/kg K], the rates $q_{e,s}$ in [kg/s], the temperatures t_e and Δt_e in [K], and the enthalpies h_e and Δh_e in [J/s] = [W].

Similar to the mass balances, we enforce an energy balance at each internal node (conjunction) in the network. With the enthalpy variables available the energy balances are easily expressed as:

$$\sum_{e \in \mathbf{E}_i^{\text{in}}} (h_e - \Delta h_e) = \sum_{e \in \mathbf{E}_i^{\text{out}}} h_e, \quad \forall i \in \mathbf{N}^{\text{int}}. \tag{8}$$

Note that the downstream enthalpies ($h_e - \Delta h_e$) are used in the left-hand side of Eq. (8) to summarize the energy entering the node. According to the assumptions, the energy balances above are correct if: 1) no work is performed, 2) no heat is added, and 3) that the net change in kinetic and potential energy is zero.

The inclusion of the above energy model may increase the accuracy of the overall problem formulation. However, the increased accuracy comes at the cost of a computational heavier formulation since $3|\mathbf{E}|$ nonconvex constraints (nonlinear equality constraints) are added to the problem. One upside with the energy model is that it does not involve any binary routing variables. To see why, consider the case when a discrete edge is closed ($y_e = 0$). The flow rate is then forced to zero, which in turn forces the enthalpy on the edge to zero. Consequently, it does not contribute to the energy balance in Eq. (8). Thus, there is no need to involve binary variables in the energy balances.

4.4. Flow routing

Flows can be routed through certain parts of the network by opening and closing discrete edges: closing a discrete edge forces its mass flow to zero. This behaviour is expressed by combining the binary variable y_e with the lower ($q_{e,s}^L$) and upper ($q_{e,s}^U$) bounds on the flow rate as:

$$y_e q_{e,s}^L \leq q_{e,s} \leq y_e q_{e,s}^U, \quad \forall s \in \mathbf{S}, e \in \mathbf{E}^{\text{d}}. \tag{9}$$

Note that Eq. (9) may force the flow rates $\mathbf{q}_e$ to zero. Thus, the domains of the nonlinear functions $f_e(\cdot)$ and $g_e(\cdot)$ should contain $\mathbf{q}_e = \mathbf{0}$; otherwise, $y_e = 0 \implies \mathbf{q}_e = \mathbf{0}$ is infeasible.

Depending on the network topology, some binary variables may be redundant. For example, the mass balance of a node with one entering and one leaving discrete edge will enforce equal flow rates. Thus, closing any one edge will force the flow rate on the other edge to zero. In this case one discrete edge (a single binary variable) is sufficient to model the on/off logic.

In general, the discrete edges may be configured to model any routing problem. Next we discuss a common routing configuration called a manifold.

4.4.1. The manifold: a special routing structure

A manifold is a collection of pipes and on/off valves designed so that its inlets can be routed to its outlets in various configurations, possibly by commingling the inlet streams. In a graph, the analogue to a manifold is a set of discrete edges connected as shown in Figure 2.

A normal operational constraint on subsea manifolds is that an inlet stream can be routed to at most one of the outlets. This constraint enforces requirement R2; that a node may have at most one open outlet. The manifold routing constraints are easily expressed with the binary variables of the discrete edges as

$$\sum_{e \in \mathbf{E}_i^{\text{out}}} y_e \leq 1, \quad \forall i \in \mathbf{N}^{\text{d}}, \tag{10}$$

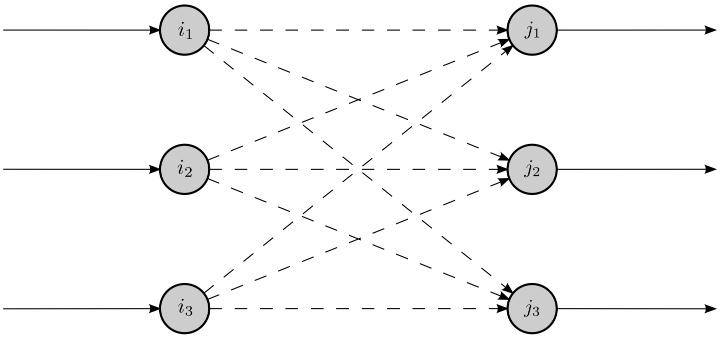

Figure 2: A manifold modelled with discrete edges (dashed lines). The manifold can route each of the three inlet streams to any of the three outlets.

where $\mathbf{N}^{\mathrm{d}}$ are nodes with discrete leaving edges.[5] The constraints in Eq. (10) allow flow to none or one of the edges leaving a node. Routing to zero edges forces the phase flow rates to zero via the mass balances in Eq. (1) and flow routing constraints in Eq. (9). If the inlet stream is required to flow to exactly *one* outlet the inequality in Eq. (10) is replaced with equality.

A manifold with 9 discrete edges configured as in Figure 2 has a total of $2^9 = 512$ possible routing combinations. The cut in Eq. (10) reduces the number of feasible routing combinations to $2^6 = 64$. A quick way to calculate the number of feasible combinations is to find the number of feasible combinations for each node, which is $n + 1$ for a node with n discrete leaving edges, and then multiply these numbers together. As calculated above we obtain $n + 1 = 4$ for all three nodes, and $4 \cdot 4 \cdot 4 = 4^3 = 2^6 = 64$ feasible combinations.

Note that a manifold is constructed to minimize pressure loss across its pipes and valves, hence the pressure drop over the discrete edges may be fixed to zero.

4.5. Boundary conditions

To obtain a well-posed flow network problem it is necessary to specify boundary conditions for the network. The boundary conditions are usually related to the source and sink nodes, and specify the interaction between the network and its neighbouring systems. Next we discuss a few upstream and downstream boundary conditions, commonly used in models of subsea petroleum production networks.

4.5.1. Upstream boundary conditions

At a source node $i \in \mathbf{N}^{\mathrm{src}}$ we assume that the following relation between the pressure p_i and flow rates $\mathbf{q}_e$ exists:

$$\zeta_{i,s}(\mathbf{q}_e, p_i) = 0, \quad \forall s \in \mathbf{S}, i \in \mathbf{N}^{\mathrm{src}}, \tag{11}$$

where $\zeta_{i,s} : \mathbb{R}^{|\mathbf{S}|+1} \to \mathbb{R}$, and $e = (i, j)$ is the (only) edge leaving source node i.

A common class of inflow rate boundary conditions in subsea production networks is the *inflow performance relationship* (IPR). It describes the mass flow from the reservoir into the

[5] The constraint in Eq. (10) is redundant for nodes with only one discrete leaving edge: the constraint would be $y_e \leq 1$, which is always true. These redundant constraints can easily be omitted by altering Eq. (10) to apply only to nodes $i \in \{j : j \in \mathbf{N}^{\mathrm{d}}, |\mathbf{E}_j^{\mathrm{out}}| > 1\}$. However, to keep the notation simple the manifold routing constraint is applied to all nodes in $\mathbf{N}^{\mathrm{d}}$.

well as a function of the measured bottom-hole pressure (also known as draw-down pressure). Wells without bottom-hole pressure sensors are usually modelled with a *well performance curve* (WPC), relating the flow rate to the wellhead pressure.

Two widely used IPRs are the linear (straight line) IPR and Vogel's quadratic IPR (Ahmed et al., 2006). With a linear IPR, a well $i \in \mathbf{N}^{\text{src}}$ can be modelled with the linear constraints:

$$\begin{aligned} q_{e,\text{oil}} &= c_{i,\text{PI}}(p_{i,\text{res}} - p_i), \\ q_{e,\text{gas}} &= c_{i,\text{GOR}} \cdot q_{e,\text{oil}}, \\ q_{e,\text{wat}} &= \frac{c_{i,\text{WCT}}}{100 - c_{i,\text{WCT}}} \cdot q_{e,\text{oil}}. \end{aligned} \tag{12}$$

There are four constants in Eq. (12) that characterizes a well: the reservoir block pressure $p_{i,\text{res}}$ (which is considered constant according to assumption A1), the productivity index $c_{i,\text{PI}}$, the gas-oil ratio $c_{i,\text{GOR}} \geq 0$, and the water cut $c_{i,\text{WCT}} \in [0, 100)$.

The linear model in Eq. (12) does not hold for all reservoirs. For instance, it does not hold for reservoirs with a thin oil rim overlaid by a large gas cap, where wells are subject to gas coning. In coning wells, the gas rate varies nonlinearly with the oil flow rate, and Eq. (12) should be substituted with nonlinear relations. These relations may be generated from near-well simulations performed by a reservoir simulator (Mjaavatten et al., 2008).

When temperatures are included in the model, it is customary to assume that the temperature of the entering fluid is constant and equal to the reservoir temperature, that is

$$t_e = \text{const.}, \quad e \in \mathbf{E}^{\text{src}}. \tag{13}$$

Reasonably accurate inflow models is a prerequisite for an accurate network model. This part of the model is, however, hard to calibrate. In practise, experiments need to be performed to collect data for inflow model calibration. This usually involves disruptive well testing, where a single well is routed to a test header to allow the measuring of flow rates over a time span of hours.

4.5.2. Downstream boundary conditions

In line with Assumption A1 it is reasonable to assume a constant downstream (separator) pressure when modelling a subsea production system. The constraints are easily expressed as:

$$p_i = \text{const.}, \quad i \in \mathbf{N}^{\text{snk}}, \tag{14}$$

where a constant pressure is specified for the sink (separator) nodes.

4.6. Operational constraints

In daily production optimization the production engineers must consider many operational constraints. To obtain solutions with practical value these constraints must be included in the optimization problem. Here we mention two very common operational constraints: namely the capacity and draw-down constraint.

4.6.1. Capacity constraints

In daily production optimization, the topside separator is typically considered to be the downstream boundary of the network. Hence, the amount of fluid entering the separator must honour the water and gas handling capacity of the downstream process facility. The capacity constraints on the total production of gas and water are easily expressed by cut sets (here we have used the set of sink edges $\mathbf{E}^{\text{snk}}$):

$$\sum_{e \in \mathbf{E}^{\text{snk}}} q_{e,\text{gas}} \leq C_{\text{gas}} \text{ and } \sum_{e \in \mathbf{E}^{\text{snk}}} q_{e,\text{wat}} \leq C_{\text{wat}}, \tag{15}$$

where the total gas (water) flowing into the separator/sink nodes is limited by the gas (water) handling capacity C_{gas} (C_{wat}).

4.6.2. Draw-down constraints

A draw-down constraint is a lower limit on the bottom hole pressure of a well. The constraint prevents operation at pressures and thereby rates that potentially can damage the well and near-well reservoir. Let $i \in \mathbf{N}$ be a node representing the bottom hole of a well. Then a draw-down constraint on i is expressed with the bounds on p_i: i.e., $p_i^L \leq p_i \leq p_i^U$, where the lower bound p_i^L specifies the draw-down limit.

4.7. Objective function

As discussed in the introduction, the main objective when optimizing a petroleum network is typically the maximization of oil production. This objective is easily expressed by summing the oil rates of all edges in a cut set. Two obvious cut sets are the edges leaving a source node ($\mathbf{E}^{\text{src}}$) or the edges entering a sink node ($\mathbf{E}^{\text{snk}}$). Below we express the objective function using the latter.

$$\text{maximize} \quad z = \sum_{e \in \mathbf{E}^{\text{snk}}} q_{e,\text{oil}}, \tag{16}$$

Sometimes it makes sense to include contributions to the cost of operating in the objective function; for example the cost of utilizing gas lift or processing produced water. In this framework it is straightforward to include these in the objective.

4.8. Flow network: a MINLP formulation

With the complete flow network modelled, the daily production optimization problem is posed as the following mixed-integer nonlinear programming problem:

$$
\begin{array}{lll}
\underset{\mathbf{y},\mathbf{q},\mathbf{p},\mathbf{\Delta p},\mathbf{t},\mathbf{\Delta t},\mathbf{h},\mathbf{\Delta h}}{\text{maximize}} & z = \displaystyle\sum_{e \in \mathbf{E}^{\text{snk}}} q_{e,\text{oil}} & \\
\text{subject to} & \displaystyle\sum_{e \in \mathbf{E}_i^{\text{in}}} q_{e,s} - \sum_{e \in \mathbf{E}_i^{\text{out}}} q_{e,s} = 0, & \forall s \in \mathbf{S}, i \in \mathbf{N}^{\text{int}} \\
& \Delta p_e = f_e(\mathbf{q}_e, p_i, t_e), & \forall e \in \mathbf{E} \backslash \mathbf{E}^{\text{d}} \\
& \Delta p_e = p_i - p_j, & \forall e \in \mathbf{E} \backslash \mathbf{E}^{\text{d}} \\
& -M_e(1-y_e) \leq p_i - p_j - \Delta p_e \leq M_e(1-y_e), & \forall e \in \mathbf{E}^{\text{d}} \\
& \Delta t_e = g_e(\mathbf{q}_e, p_i, t_e), & \forall e \in \mathbf{E} \\
& h_e = t_e \displaystyle\sum_{s \in \mathbf{S}} c_s \cdot q_{e,s}, & \forall e \in \mathbf{E} \\
& \Delta h_e = \Delta t_e \displaystyle\sum_{s \in \mathbf{S}} c_s \cdot q_{e,s}, & \forall e \in \mathbf{E} \\
& \displaystyle\sum_{e \in \mathbf{E}_i^{\text{in}}} (h_e - \Delta h_e) = \sum_{e \in \mathbf{E}_i^{\text{out}}} h_e, & \forall i \in \mathbf{N}^{\text{int}} \qquad (\mathbf{P}) \\
& \displaystyle\sum_{e \in \mathbf{E}_i^{\text{out}}} y_e \leq 1, & \forall i \in \mathbf{N}^{\text{d}} \\
& y_e q_{e,s}^L \leq q_{e,s} \leq y_e q_{e,s}^U, & \forall s \in \mathbf{S}, e \in \mathbf{E}^{\text{d}} \\
& q_{e,s}^L \leq q_{e,s} \leq q_{e,s}^U, & \forall s \in \mathbf{S}, e \in \mathbf{E} \backslash \mathbf{E}^{\text{d}} \\
& p_i^L \leq p_i \leq p_i^U, & \forall i \in \mathbf{N} \\
& t_e^L \leq t_e \leq t_e^U, & \forall e \in \mathbf{E} \\
& \zeta_{i,s}(\mathbf{q}_e, p_i) = 0, & \forall s \in \mathbf{S}, i \in \mathbf{N}^{\text{src}} \\
& p_i = \text{const.}, & \forall i \in \mathbf{N}^{\text{snk}} \\
& t_e = \text{const.}, & \forall e \in \mathbf{E}^{\text{src}} \\
& y_e \in \{0,1\}, & \forall e \in \mathbf{E}^{\text{d}}
\end{array}
$$

In the rest of this work we denote an optimal value of **P** by z^*, obtained at an optimal solution $(\mathbf{x}^*, \mathbf{y}^*)$, where $\mathbf{x}$ is a vector containing all the continuous variables in **P**. Notice that the problem is nonconvex due to the integer variables and the nonlinear equality constraints Eqs. (2), (6), (7), and (11). Consequently, we cannot expect to find a global optimum, unless the problem is solved with a global solver.

The formulation in **P** can be used to model any nonlinear flow network under Assumptions A1-A4, and topology requirements R1-R3. For problems not requiring an accurate energy model, a cruder model can be obtained by removing from **P** the temperature and enthalpy variables, as well as the constraints for energy conservation. This will remove $3|\mathbf{E}|$ nonconvex constraints, simplifying the problem considerably. For a subsea production network, the framework allows for modelling of gas lifted wells and complex multi-branch wells by the addition of nodes and edges.

A key property of **P**, which may not present itself immediately, is that the integer variables participate in linear constraints only. This is an advantageous property since the discrete logic may be exclusively handled by the solver. In some cases the nonlinearities are represented by process simulators without the capacity to handle discrete logic.

Another important aspect of the formulation is that it does not contain functions of more than $|\mathbf{S}| + 2$ variables (rates, pressure, and temperature). This allows the nonlinear functions to be replaced with approximations/surrogates of low dimension. In Section 5 we show how the nonlinear functions $f(\cdot)$ may be approximated with spline surrogate models. As will become clear later in Section 6, this allows us to solve **P** to global optimality with a spline-compatible solver.

Before continuing, we would like to remark on the fact that choke openings are not directly computed in **P**. Chokes are usually modelled with nonlinear Cv curves, relating the choke opening and the differential pressure over the choke. To avoid the additional nonlinearity of the Cv curves when optimizing, the choke are represented by a differential pressure variable (Δp_e). The choke openings are back-calculated from the optimal differential pressures after solving **P**.

5. Spline surrogate models

In this section we give a brief introduction to function approximation with splines. Our purpose is to motivate the use of splines as surrogates for the nonlinear functions in the optimization problem, **P**. We will use a light notation and represent the splines as basis splines, or *B-splines*. For a detailed treatment of B-splines we refer the reader to the literature on spline theory; cf. the textbooks of Piegl and Tiller (1997) and Schumaker (2007).

5.1. Univariate and multivariate B-splines

A spline is a piecewise polynomial function which possesses a required degree of smoothness at the points where the polynomial pieces connect (which are called *knots*). First we consider the univariate B-spline, denoted as

$$\phi_p(x) = \mathbf{c}^\mathsf{T}\mathbf{b}_p(x), \tag{17}$$

where $\mathbf{c} \in \mathbb{R}^n$ is a vector of n coefficients and $\mathbf{b}_p \in \mathbb{R}^n$ is a vector of n *B-spline basis functions*. The basis functions in $\mathbf{b}_p$ are (overlapping) p-th degree polynomial pieces in the variables x; see Figure 3 for an illustration. They are recursively constructed from a nondecreasing sequence of $n + p + 1$ real numbers $t_1 \leq \ldots \leq t_{n+p+1}$ known as *knots*. These numbers are often collected in a vector $\mathbf{t} = \{t_i\}_{i=1}^{n+p+1}$, called the *knot vector*. Note that with our notation the dependence of $\mathbf{b}_p$, and ϕ_p, on $\mathbf{t}$ is implied. We refer the reader to the literature for a description of the relation between the knots and the basis functions.

The B-spline ϕ_p is a linear combination of basis functions and consequently a piecewise polynomial with degree p. An important property of the B-spline is that it has local support, meaning that at most $p + 1$ basis functions are nonzero at a point x. This, in addition to several other advantageous properties, allow fast and numerically stable methods for manipulation and evaluation of splines; see for example De Boor (1972) and Cox (1972).

The B-spline generalizes nicely to the multivariate case. A degree p B-spline in the variables $\mathbf{x} \in \mathbb{R}^d$ may be compactly written as

$$\phi_p(\mathbf{x}) = \mathbf{c}^\mathsf{T}\mathbf{B}_p(\mathbf{x}), \tag{18}$$

where $\mathbf{B}_p \in \mathbb{R}^N$ is a vector of N multivariate B-spline basis functions of degree p. The multivariate basis functions are obtained by taking the tensor product of univariate basis functions,

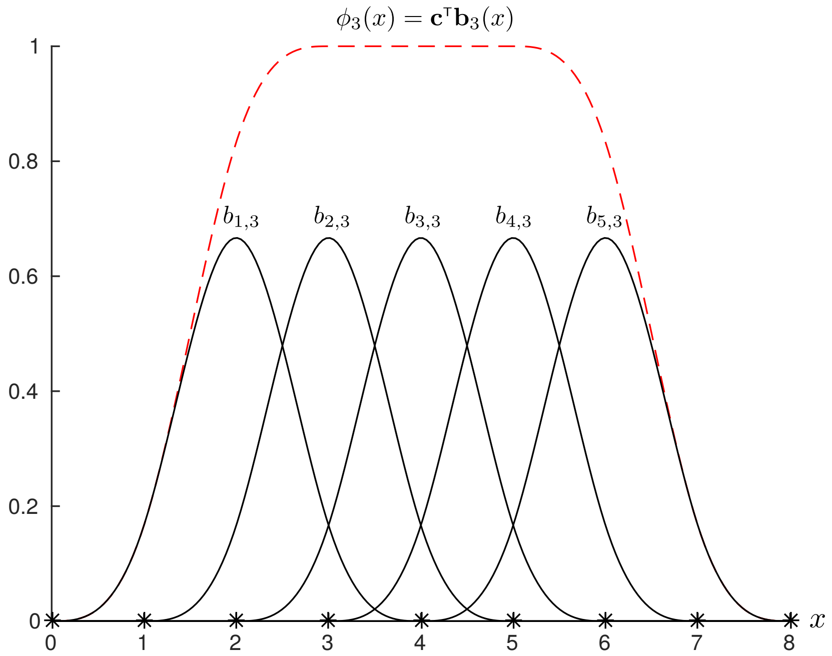

Figure 3: Illustration of the univariate, cubic B-spline basis functions $\mathbf{b}_3 = [b_{i,3}]_{i=1}^5$ for the knot vector $\mathbf{t} = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$ (marked with asterisks on the x-axis). The cubic B-spline $\phi_3(x) = \mathbf{c}^\mathsf{T}\mathbf{b}_3(x)$, with coefficients $\mathbf{c}^\mathsf{T} = [11111]$, is also shown.

i.e.

$$\mathbf{B}_p(\mathbf{x}) = \mathbf{b}_p^1(x_1) \otimes \ldots \otimes \mathbf{b}_p^d(x_d) = \bigotimes_{i=1}^{d} \mathbf{b}_p^i(x_i), \tag{19}$$

where $\otimes$ denotes the Kronecker product.[6] The Kronecker product produces a vector $\mathbf{B}_p$ that contains all possible combinations of the univariate bases: this results in a total of $N = n_1 \cdots n_d$ multivariate basis functions, where n_i is the number of univariate basis functions in $\mathbf{b}_p^i(x_i)$ in variable x_i. Each basis function vector $\mathbf{b}_p^i$ in Eq. (19) is parametrized by its own knot vector $\mathbf{t}_i$. Note that a multivariate basis function is a product of d degree p univariate basis functions, making it a multivariate, piecewise polynomial of degree dp.[7] The domain of $\phi(\mathbf{x})$ is considered to be the box $X = X_1 \times \cdots \times X_d$, where X_i is the interval supported by at least one basis function in $\mathbf{b}_p^i(x_i)$. Consequently, $\phi(\mathbf{x}) = 0, \forall \mathbf{x} \notin X$.

Most properties of the univariate B-spline carry over to the multivariate case. For example, the multivariate B-spline also enjoys local support and have fast algorithms for manipulation and evaluation (although their implementation require extra care to exploit sparsity patterns). The multivariate B-spline is a powerful modelling and approximation tool, as is testified by the numerous computer-aided design tools that use it. A broad application follows from the fact

[6] In the literature the multivariate B-spline is often referred to as tensor product B-spline since the basis functions are constructed using the tensor product.

[7] To ease the notation in Eqs. (18) and (19) we have assumed that all univariate basis functions vectors $\mathbf{b}_p^i$ are of the same degree p. This assumption can easily be removed without any consequences.

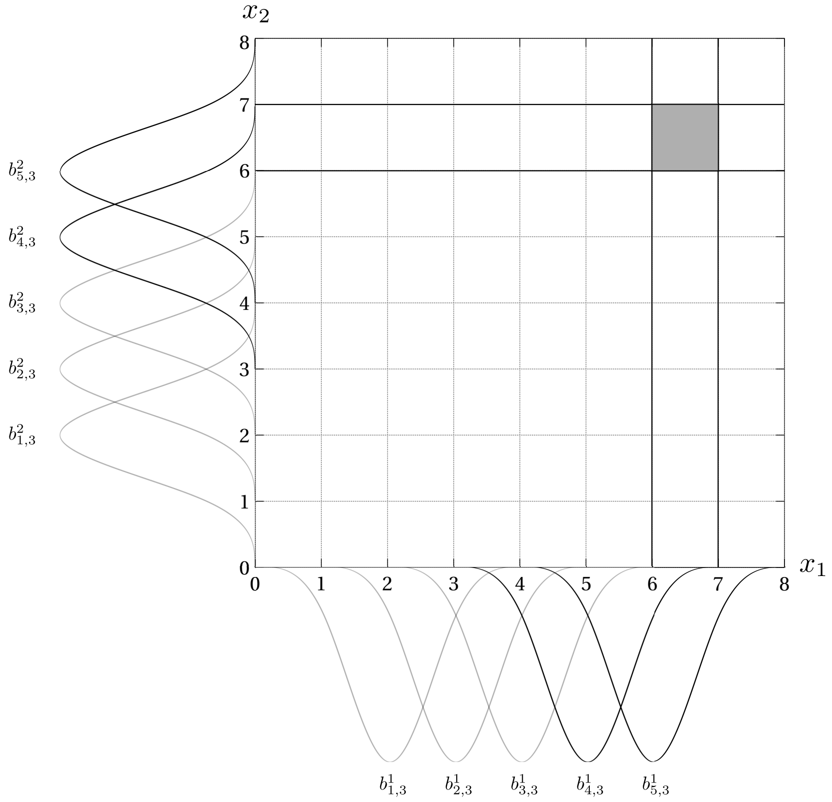

Figure 4: Illustration of the univariate, cubic B-spline basis functions $\mathbf{b}^1_3 = [b^1_{i,3}]^5_{i=1}$ and $\mathbf{b}^2_3 = [b^2_{i,3}]^5_{i=1}$ for the knot vectors $\mathbf{t}_1 = \mathbf{t}_2 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. Bivariate, cubic basis functions are constructed by $\mathbf{B}_p = \mathbf{b}^1_3 \otimes \mathbf{b}^2_3$. The grey box $\mathbf{x} \in [6, 7]^2$ is supported by the following bivariate basis functions: $[b^1_{4,3} b^1_{4,3}] \otimes [b^1_{4,3} b^1_{4,3}] = [b^1_{4,3} b^2_{4,3}, b^1_{4,3} b^2_{4,3}, b^1_{5,3} b^2_{4,3}, b^1_{5,3} b^2_{4,3},]$.

that the B-spline may represent any piecewise polynomial function exactly, that is, without any approximation error. Models containing non-polynomial functions, such as the transcendental functions, may only be approximated by a B-spline. The approximation error can then be controlled by changing the density of the samples. In the next section we show how to approximate a function that has been sampled on a grid with a B-spline.

5.2. Function approximation with B-splines

Let any function $f : \mathbb{R}^d \to \mathbb{R}$ be sampled on a regular grid to yield m data points $\{\mathbf{x}^i, y^i\}^m_{i=1}$, where $f(\mathbf{x}^i) = y^i$. Using only these data points a B-spline that approximates f is constructed. Several approximation methods exist and they are usually categorized as being interpolating or smoothing. Among the interpolating methods the widely used cubic spline interpolation is most common. There are especially three reasons for the popularity of cubic spline interpolation: 1) it is fast to compute, 2) it offers a high degree of smoothness, and 3) it is a good approximation to a broad class of functions.

An interpolation method computes a B-spline that interpolates f at all of the m data points.

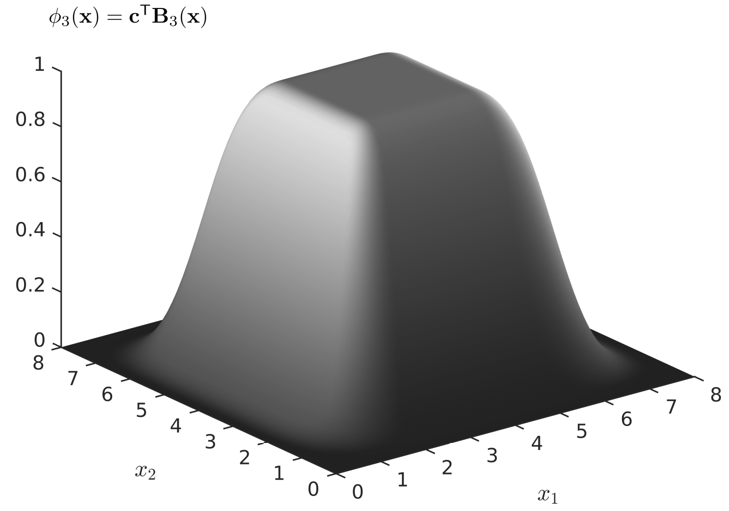

Figure 5: A bivariate, cubic B-spline constructed from the basis functions in Figure 4 and the coefficients $\mathbf{c} = \mathbf{1}_{25}$, where $\mathbf{1}_{25} \in \mathbb{R}^{25}$ is a vector of $5 \times 5 = 25$ ones.

Mathematically, the following linear system is solved for the coefficients $\mathbf{c}$:

$$\underbrace{\Big[\mathbf{B}_p(\mathbf{x}^1)\ \mathbf{B}_p(\mathbf{x}^2)\ \ldots\ \mathbf{B}_p(\mathbf{x}^m)\Big]^\mathsf{T}}_{\mathsf{B}_c} \mathbf{c} = \mathbf{y} \tag{20}$$

In Eq. (20) $\mathbf{y} = [y^i]_{i=1}^m$ and $\mathsf{B}_c \in \mathbb{R}^{m \times N}$ is the so-called *B-spline collocation matrix*: the matrix where row i corresponds to the vector of basis functions $\mathbf{B}_p(\mathbf{x}^i)$ evaluated at sample $\mathbf{x}^i$. It is customary to select a knot vector that gives a square collocation matrix ($N = m$). An example of such a knot vector is the *free end conditions* knot vector for cubic spline interpolation ($p = 3$):

$$\mathbf{t}_F = \{\underbrace{x^1, \ldots, x^1}_{p+1 \text{ repetitions}}, x^3, \ldots, x^{m-2}, \underbrace{x^m, \ldots, x^m}_{p+1 \text{ repetitions}}\}.$$

Notice that the second and second last knot is omitted from $\mathbf{t}_F$ to give $N = m$. For square B_c, the conditions under which B_c is invertible are known as the *Schoenberg-Whitney nesting conditions*: $t_i < x^i < t_{i+p+1}$ for $i = 1, 2, \ldots, m$, allowing $x^i = t_i$ only if $t_i = t_{i+p} < t_{i+p+1}$. These conditions are fulfilled for $\mathbf{t} = \mathbf{t}_F$. When B_c is square and invertible, the B-spline coefficients can readily be computed by solving $\mathsf{B}_c\mathbf{c} = \mathbf{y}$.

The $m \times m$ linear system in Eq. (20) can be solved efficiently by a sparse solver on a modern desktop computer for $m \leq 100,000$. For example, when approximating a function in 5 variables this practical limit allows a discretization with 10 values in each variables, resulting in a grid of $10^5 = 100,000$ sample points. For a "well-behaving", low-dimensional function such as a well lift curve, 100,000 samples is more than enough to achieve an accurate approximation. In this work we have utilized the SPLINTER library for function approximation (Grimstad et al., 2015b).

Before we illustrate cubic spline interpolation in the next subsection, we point out that the approximation error of a spline can be made arbitrarily small for continuous functions by increasing the sampling density. Furthermore, the approximation error is dependant on the knot placement. Optimal knot placement, however, is a difficult and largely unresolved problem, particularly for multivariate B-splines (Natali and Pinto, 2009). Luckily, when samples are taken on a regular grid the knots can be set equal to the sample points as in $\mathbf{t}_F$.[8] With scattered (irregular) sample points it is not trivial to select the knot vectors. In either case, the problem of where to place the sample points still remains and is highly dependant on the function to be approximated.

5.3. An example: Beggs and Brill approximated with a B-spline

Figure 6 shows the Beggs and Brill pressure drop correlation for a slightly inclined pipe. With the given parameters, the correlation includes three different flow regimes on the domain. By inspecting the figure one may observe several bends in the correlation; the groove between the segregated and transition flow regime is conspicuous.

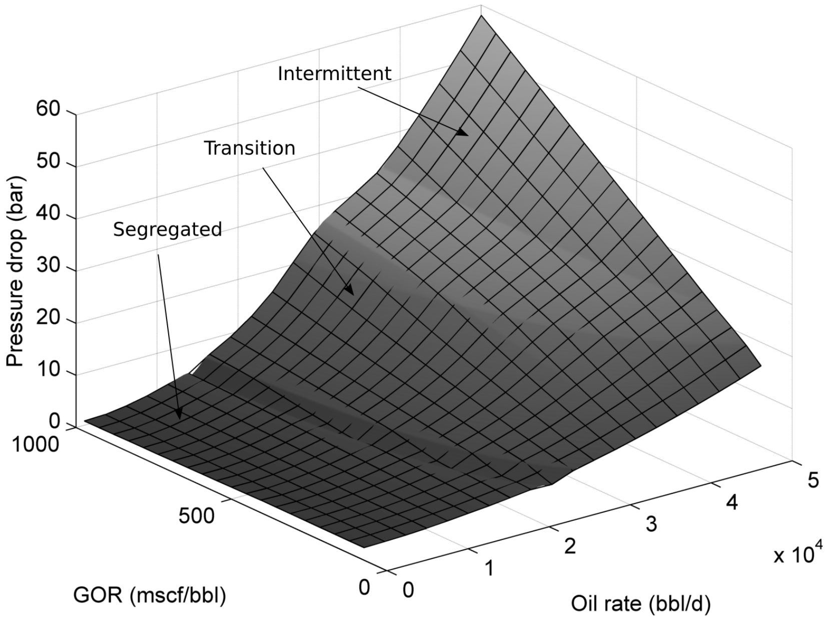

Figure 6: Beggs and Brill pressure drop correlation for a 1 degree inclined, 1000 meter long, 12 inch pipe. The water-cut is fixed at 10% and the outlet pressure is 30 bar. As indicated, three different flow regimes occur as the oil rate increases. The overlaying grid shows the sample points used to build the spline approximations.

The 20x20 grid in Figure 6 shows the $m = 400$ points where the correlation was sampled. From these points two approximations are constructed: a linear spline ($p = 1$) and a cubic spline ($p = 3$). The approximation error of the two splines are plotted in Figure 7.

By inspecting the figure we see that qualitatively the approximation errors of the two interpolating splines are similar. The error increases along the diagonal grooves/bends where the

[8] By default, SPLINTER computes the knots by applying a moving average filter with window size $p + 2$ to the sample points. With equidistant samples this filter produces the knot vector $\mathbf{t}_F$.

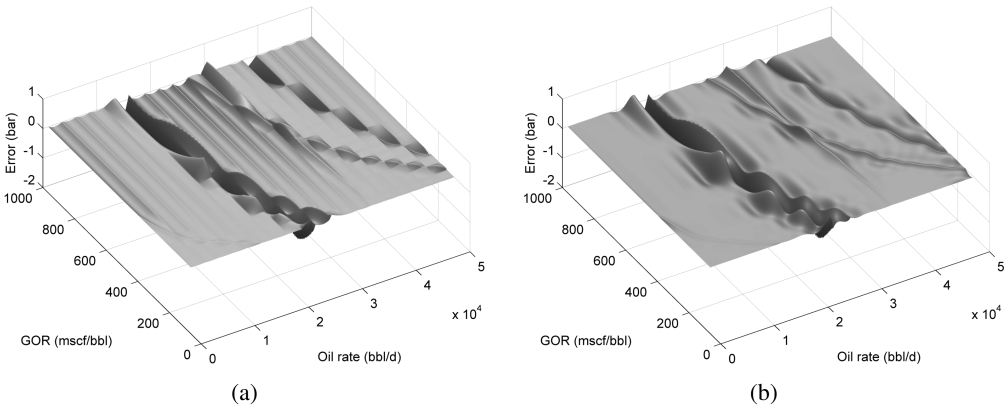

Figure 7: The approximation error of the linear spline (a) and the cubic spline (b) interpolating the Beggs and Brill pressure drop correlation on the equidistant 20x20 grid in Figure 6.

rectangular grid fails to capture the geometry. The most notable difference between the two splines are the ripples in Figure 7a. The ripples show that the linear spline fails to capture the curvature between the grid lines; this effect is not present to the same degree in Figure 7b.

To quantitatively compare the interpolating splines we measure the relative approximation error with $\epsilon_{\text{rel},2} = \|1 - \phi(\mathbf{x})/f(\mathbf{x})\|_{2,X}$ and $\epsilon_{\text{abs},\infty} = \|f(\mathbf{x}) - \phi(\mathbf{x})\|_{\infty,X}$ for $\mathbf{x} \in X$, where $\|\cdot\|_{p,X}$ denotes the L^p-norm on the domain X. The errors for the two splines are given in Table 5.

Table 5: Spline approximation errors.

	$\epsilon_{\text{rel},2}$	$\epsilon_{\text{abs},\infty}$
Linear spline	0.0044	1.1193
Cubic spline	0.0039	1.0545

6. Solution method

In this section we present the proposed method for solving **P**. The main assumption is that *all* the nonlinear functions in **P** are B-spline functions on the form in Eq. (18), that is, we assume that $f_e(\cdot)$, $g_e(\cdot)$, and $\zeta_{i,s}(\mathbf{q}_e, p_i)$ are B-spline functions. The bilinear terms in the enthalpy constraints in Eq. (7) may also be represented exactly with splines.[9] With this assumption problem **P** falls into the category of *spline-constrained* MINLP problems, which may be solved to global optimality by the spline-compatible optimization framework CENSO (Convex ENvelopes for Spline Optimization), recently presented by Grimstad and Sandnes (2014). CENSO is publicly available as open-source C++ code (Grimstad et al., 2015a). A description of the algorithm is given in the next section. Please note that without loss of generality we assume **P** to be a minimization problem in this description. After the description, we present two improvements that may speed up the algorithm when solving production optimization problems.

[9] A spline may represent a bilinear term exactly. In fact, the convex hull relaxation of the B-spline is identical to the McCormick relaxation for bilinear terms (Grimstad and Sandnes, 2014).

6.1. *Description of* CENSO

CENSO is a framework for optimization with spline constraints. It contains a spatial branch-and-bound (sBB) algorithm that partitions the problem domain by branching on continuous variables as well as integer variables; this produces subproblems $\mathbf{P}_k$ of $\mathbf{P}$. Spline (equality) constraints are generally nonlinear, thus non-convex, and must be relaxed during the solution process. CENSO employs lifted polyhedral sets to relax spline constraints, producing relaxed LP subproblems, denoted $\mathbf{R}_k$.

Let $\hat{z}_k$ denote the solution to $\mathbf{P}_k$ and $\bar{z}_k$ the solution to $\mathbf{R}_k$. The fact that $\hat{z}_k \geq \bar{z}_k$ is used to process the search domain. The current best feasible solution found, known as the incumbent, is denoted z^u. The algorithm described next, terminates when it has proved that there cannot exist a solution better than $z^u - \epsilon$ (where $\epsilon > 0$ is a small number). This is known as ϵ-convergence.

The schematic in Figure 8 describes the sBB algorithm in CENSO. From the top: the algorithm is initialized by adding $\mathbf{P}$ to the list of problems $\mathcal{L}$, and setting the upper bound on $\mathbf{P}$ to $z^u = \infty$. The algorithm then enters a loop which terminates when $\mathcal{L}$ is empty. Upon termination there are two possible outcomes: a global optimum has been found (ϵ-convergence) or the problem is infeasible.

The first step inside the loop is to select and remove from $\mathcal{L}$ the next subproblem $\mathbf{P}_k$ to be processed. The sBB in CENSO uses a simple *best-bound-first* policy, selecting the subproblem with the lowest lower bound $\bar{z}_k$ (inherited from its parent node).

After selection, bounds tightening techniques are applied to $\mathbf{P}_k$. The purpose of these techniques is to reduce the domain of $\mathbf{P}_k$ and hence to accelerate the exploration of the search space. These methods may also prove $\mathbf{P}_k$ infeasible, in which case it is fathomed.

Next, the convex relaxation $\mathbf{R}_k$ is generated and solved to get $\bar{z}_k$. By default, the convex relaxation is solved by Gurobi (Gurobi Optimization, Inc., 2014). With the lower bound on $\mathbf{P}_k$ three fathoming rules are checked: (i) $\bar{z}_k \geq z^u$, (ii) $\bar{z}_k = \infty$ ($\mathbf{R}_k$ infeasible), and (iii) $z^u - \bar{z}_k \leq \epsilon$ (converged). If any of (i)–(iii) are true, the node is fathomed as it may not contain a solution better than $z^u - \epsilon$.

If the subproblem cannot be fathomed its domain needs further processing. First, the incumbent is updated by checking if the solution to $\mathbf{R}_k$ is feasible to $\mathbf{P}_k$. To further improve the incumbent an NLP or MINLP solver may be used to find a feasible solution of $\mathbf{P}_k$ that is better than z^u. This heuristic is not required, but may speed up the convergence of the search.

Finally, at the end of the loop a continuous or integer branching variable is selected for $\mathbf{P}_k$. This variable is then branched on to create two new partitions $\mathbf{P}_{k-}$ and $\mathbf{P}_{k+}$. The two partitions are added to the list $\mathcal{L}$, completing one iteration of the loop. Note that after one loop iteration the list size $|\mathcal{L}|$ is either decremented by one (if $\mathbf{P}_k$ fathomed) or incremented by one (if $\mathbf{P}_k$ is branched on). If the list is empty, the search terminates with the optimal solution z^u.

6.2. *Branching variables and bounds tightening*

A requirement for the sBB algorithm to converge to a global optimum is that it may branch on all *complicating variables*. In a MINLP problem, the complicating variables are the integer variables and any continuous variable that participates *nonlinearly* in a nonconvex constraint. For problem $\mathbf{P}$, the complicating variables are $\mathbf{y}$ and $\mathbf{x}_c$, where $\mathbf{x}_c = [x_i]_{i \in \mathcal{I}_c}$ are the complicating continuous variables. The index set $\mathcal{I}_c$ is given so that $\{x_i : i \in \mathcal{I}_c\} = \{\mathbf{q}_e, p_i, t_e, \Delta t_e : e = (i, j) \in \mathbf{E}\}$. Note that $\mathbf{x}_c$ does not contain the variables Δp_e, h_e, and Δh_e since they participate linearly in all constraints (the reader can verify this by looking at $\mathbf{P}$).

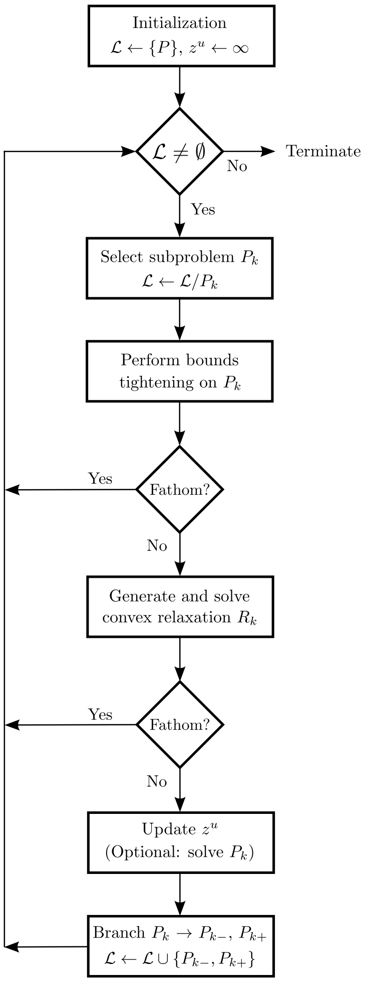
Initialization
$\mathcal{L} \leftarrow \{P\}$, $z^u \leftarrow \infty$
$\mathcal{L} \neq \emptyset$
No
Terminate
Yes
Select subproblem P_k
$\mathcal{L} \leftarrow \mathcal{L}/P_k$
Perform bounds tightening on P_k
Yes
Fathom?
No
Generate and solve convex relaxation R_k
Yes
Fathom?
No
Update z^u
(Optional: solve P_k)
Branch $P_k \rightarrow P_{k-}, P_{k+}$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{P_{k-}, P_{k+}\}$

Figure 8: The spatial bound-and-bound algorithm in CENSO.

A continuous branching variable, like an integer branching variable, must be branched on a finite number of times to ensure ϵ-convergence. However, the required number of branches may be large depending on the value of ϵ and the convergence rate of the convex relaxations. At any rate, it is highly desirable to keep the number of continuous branching variables at a minimum.

Problem **P** has a relatively large number of nonconvex constraints and, as a result, $\mathbf{x}_c$ contains most of the continuous variables. From computational experience, we know that branching on all of the variables in $\mathbf{x}_c$ is detrimental to the efficiency of the algorithm, even for small network problems. To alleviate the computational load we employ so-called *bounds tightening*.[10]

Bounds tightening (BT) are techniques that reduce the variable bounds $[\mathbf{x}^L, \mathbf{x}^U]$ of a problem without removing its optimal point. BT techniques with this property are said to be *valid* or to

[10] Techniques that use the constraints to reduce the variable bounds have several names in the literature, including: bounds tightening, range reduction, and interval analysis.

produce *valid inequalities*. BT will shrink the feasible set of the primal problem and its convex relaxation. In some cases it may also prove a problem infeasible. All BT techniques utilize the constraints to, in some way, reduce the variable bounds.

Let us illustrate the advantage of bounds tightening with a simple example: Let $x_1 \in [x_1^L, x_1^U]$ and $x_2 \in [x_2^L, x_2^U]$ be continuous branching variables, related via the constraint $x_1 - x_2 = 0$. Bounds tightening will propagate the variable bounds through the constraint and ensure that $x_1, x_2 \in [\max\{x_1^L, x_2^L\}, \min\{x_1^U, x_2^U\}]$. Thus, when bounds tightening is applied, it is sufficient to branch on one of the two variables: e.g. branching on x_1 will reduce the feasible range of x_2, and vice versa.

Immediately, we understand that if we branch on the variables associated with the degree of freedom in $\mathbf{P}$, bounds tightening will ensure diminishing bounds on the remaining (branching) variables. Let $\bar{\mathbf{x}}_c = [p_i]_{e=(i,j)\in\mathbf{E}^{\mathrm{d}}}$ be the $|\mathbf{E}^{\mathrm{d}}|$ free, continuous variables in $\mathbf{P}$ (according to the DOF analysis in Appendix A). Then, it is sufficient to branch on $\mathbf{y}$ and $\bar{\mathbf{x}}_c$. It is clear that $\bar{\mathbf{x}}_c \in \mathbb{R}^{|\mathbf{E}^{\mathrm{d}}|} \subset \mathbf{x}_c \in \mathbb{R}^{(|\mathbf{S}|+3)|\mathbf{E}|}$. To be more precise, with bounds tightening the number of continuous branching variables is reduced from $(|\mathbf{S}|+3)|\mathbf{E}|$ to $|\mathbf{E}^{\mathrm{d}}|$. The reduction in the number of continuous branching variables limits the tree size and accelerates the solution time of the sBB algorithm, even when accounting for the additional computational load of the bounds tightening techniques. Next, we briefly describe the bounds tightening capabilities of CENSO.

6.2.1. *Bounds tightening techniques in* CENSO

CENSO employs the following BT techniques for MINLP problems: the *reduced-cost* BT (RCBT), originally introduced for MILP problems (Ryoo and Sahinidis, 1996; Belotti et al., 2009); and*feasibility-based* BT (FBBT) (Messine, 2004; Belotti et al., 2010). These are computationally cheap techniques that perform tightening by propagating variable bounds through the constraints. They can be solved at any node in the sBB tree, but generally produce shallow cuts. To improve upon the BT capabilities of CENSO we implement the optimality-based BT (OBBT) technique used by Zamora and Grossmann (1999) and Sahinidis (2003).

With OBBT a relaxed problem $\mathbf{R}_k$ is solved with the objective to minimize or maximize one variable. This is done for each complicating variable in $\mathbf{x}_c$. Let $F(\mathbf{R}_k)$ represent the (convex) feasible region of $\mathbf{R}_k$. Then the convex problems solved by the OBBT are

$$\underset{\mathbf{x}}{\text{minimize}} \ \{\pm x_i : \mathbf{x} \in F(\mathbf{R}_k),\ z \le z^u\}, i \in \mathcal{I}_c. \tag{21}$$

Let $\tilde{x}_i^L$ and $\tilde{x}_i^U$ be the solutions for the two objectives in Eq. (21) for variable x_i. Then the new bounds on x_i are $[x_i^L, x_i^U] \cap [\tilde{x}_i^L, \tilde{x}_i^U]$. If any of the problems in Eq. (21) are infeasible, problem $\mathbf{R}_k$ must be infeasible and can therefore be fathomed.

The OBBT requires the solution of $2|\mathcal{I}_c|$ convex NLPs or LPs; when all nonlinearities of $\mathbf{P}$ are represented with B-splines the relaxed problems $\mathbf{R}_k$ are LP problems. The OBBT may be run iteratively to achieve a greater tightening of the bounds: tighter variable bounds produce tighter convex relaxations, which in turn produce tighter variable bounds. Running OBBT iteratively is expensive and yields diminishing returns. However, it may greatly reduce the size of the BB tree. Therefore, it is typically used on every subproblem down to a certain depth, and to a limited extent deeper in the BB tree.

6.3. *Primal heuristic*

At the root node of the sBB tree, the MINLP solver BONMIN (Bonami et al., 2008) is evoked to search for a feasible solution to $\mathbf{P}$. If successful, the (primal) feasible solution, being an upper bound on the solution of **P**, may help in cutting large portions of the sBB tree. BONMIN is a heuristic in this setting since it is used to find a local optimum to the nonconvex problem **P**.

7. Case studies

In this section we present a benchmark study of the solution methods in Table 6. The study includes three realistic production optimization cases from two BP operated subsea production systems, referred to as BP subsea production system 1 and 2, from here on. Note that these cases do not necessarily correspond to the normal operation of the production system. The cases are based on models implemented in the GAP software from Petroleum Experts (Petroleum Experts Ltd., 2014). The GAP models serve as reference models when comparing solutions generated from the four different solution methods.

The four solution methods in Table 6 may be described as follows. The first is a traditional approach where a proprietary gradient-based multi-start NLP solver treats the GAP model as a black-box model, calculating gradients by finite differences. The three other methods formulate problem **P** by approximating the nonlinear relations in the production system models with splines. Problem **P** is then solved using IPOPT, BONMIN, and CENSO, respectively. The branch-and-bound-based MINLP solvers BONMIN and CENSO may solve problem **P** with discrete edges (discrete variables). The proprietary solver and IPOPT, being NLP solvers, cannot handle discrete variables.

Table 6: Solution methods.

Solver	Type	Routing	Global	Model
Proprietary solver	NLP	No	No	GAP
IPOPT (Wächter and Biegler, 2006)	NLP	No	No	**P**
BONMIN (Bonami et al., 2008)	MINLP*	Yes	No	**P**
CENSO (Grimstad and Sandnes, 2014)	MINLP**	Yes	Yes	**P**

*Convex MINLPs, **Spline constrained MINLPs.

CENSO solves problem **P** to global optimality and provides an optimality certificate with the solution, i.e. the optimality gap is less than ϵ upon termination. Solving a MINLP problem to global optimality is considerably harder, and more time consuming, than attempting a local solve. To illustrate the difference, the cases were solved to local optimality using BONMIN. To improve BONMIN's chances of finding good solutions of the nonconvex problems it was configured with the following options: algorithm set to "B-BB" (standard branch-and-bound mode), num_resolve_at_root set to 10, and num_resolve_at_node set to 2. This allows BONMIN to solve the nonconvex subproblems in the BB tree from several starting points; the starting points are naively drawn from a uniform distribution limited by the variable bounds. All other options were left at their default values.

There are a few key differences between the solution methods described above that complicates comparison of the methods. First of all, the proprietary solver and IPOPT cannot handle discrete variables. Thus, we include them in the comparison only when all discrete decisions are fixed. Second, the proprietary method solves a different model/optimization problem than the other methods since it uses the GAP model directly. To achieve a somewhat fair comparison, the optimal solutions are compared by evaluating GAP at the optimal valve settings.

The three last methods in Table 6 were run on a laptop computer equipped with an Intel 2.7 GHz dual-core processor and 8 GB of RAM memory. The proprietary solver was run on another computer with favourable performance.

Note that the reported solution times do not include the time it took to build the splines in **P**. The timings for building the splines are reported at the end of this section.

7.1. Case 1: Production optimization of BP subsea production system 1

In this case we consider the subsea production system depicted in Figure 9. The production system consists of 10 wells, 4 daisy-chained manifolds, 4 flowlines and 1 riser. The system is modelled with three fluid phases, i.e. $\mathbf{S} = \{\text{oil}, \text{gas}, \text{wat}\}$.

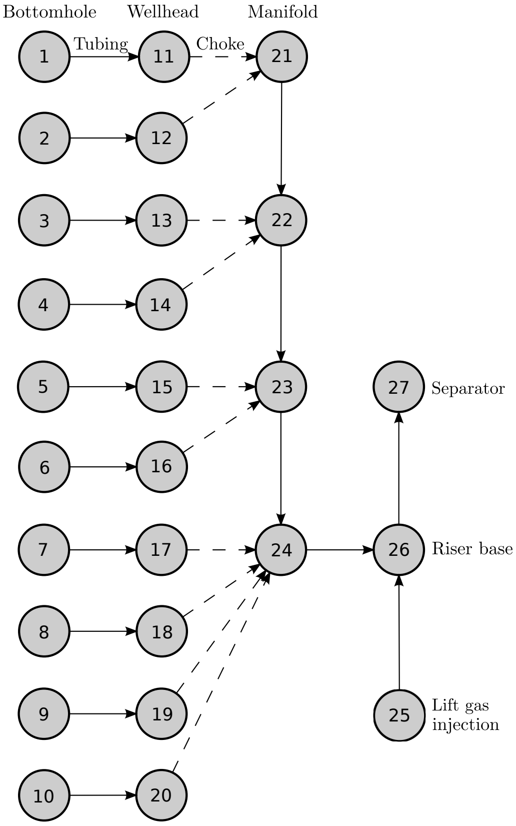

Figure 9: Topology of BP subsea production system 1.

As shown in the figure (Node 25), lift gas can be injected into the riser base to increase production by lowering the density of the fluid column. This is achieved by modelling Node 25 with the rate boundary conditions $q_{e,\text{oil}} = 0$ and $q_{e,\text{wat}} = 0$. The amount of lift gas injected into the riser is given by $q_{e,\text{gas}} \in [0, 20]$ mmscf/d. To simplify the model, the lift gas is assumed to have the same composition as the produced gas. The total gas production is limited to 340 mmscf/d, which is the gas handling capacity of the downstream processing facilities.

The wells are modelled by connecting three nodes $\{i, j, k\}$ with two edges, (i, j) and (j, k). In this configuration, depicted in Figure 10, the nodes are labeled as follows: bottomhole (i),

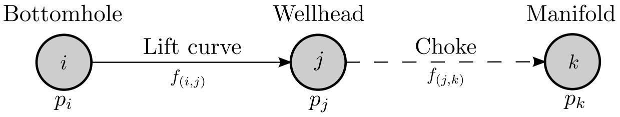

Figure 10: A well modelled with three nodes and two edges.

wellhead (j), and manifold (k). The bottomhole node (i) is a source node with no incoming edges. The inflow from the reservoir to the well is modelled by a nonlinear IPR (a piecewise function composed of a straight line and Vogel's equation), with a fixed GOR and WCT. The well parameters are listed in Table 7. The pressure drop over the wellbore, edge (i, j), is described by a nonlinear lift curve $f_{(i,j)}$ relating the flow rates $\mathbf{q}_{(i,j)}$ to the wellhead pressure p_j. The choke is described by the edge (j, k), with a related pressure drop $\Delta p_{(j,k)}$.

Table 7: Well parameters.

Well	PI (rank #)	GOR (scf/STB)	WCT (%)	p_i^L (bara)
1	6	1100	15	190
2	3	800	25	200
3	4	800	40	110
4	1	800	55	120
5	10	600	55	120
6*	2	700	50	0
7*	8	700	25	0
8	5	700	30	210
9	7	700	0	210
10	9	800	0	170

* Well is offline. ** Values are rounded for commercial reasons.

The system is modelled without any energy considerations, i.e., temperature and enthalpy variables, and related constraints, are not included in the problem formulation **P**. This reduces the number of nonconvex constraints and (complicating) variables, and hence the complexity of the problem.

We divide Case 1 into two parts. In Case 1.1 and Case 1.2 the nonlinearities are represented with linear and cubic interpolating splines, respectively. The interpolating splines are constructed by solving Eq. (20), with degree $p = 1$ for Case 1.1 and $p = 3$ for Case 1.2. Since the case includes discrete edges (binary variables) it is only solved with BONMIN and CENSO.

The results for the cases are reported in Table 8. Evidently, the cases are solved efficiently by both solvers. The solution time of CENSO is strictly higher than that of BONMIN, as is it must be since it runs BONMIN as a primal heuristic. The number of iterations used by CENSO is kept low by intensive bounds tightening, while BONMIN terminates with the global optimum after examining the root node only; i.e. it uses 0 iterations in both cases. However, BONMIN does not terminate with an optimality certificate, like CENSO.

Notice that the optimal value of the two cases differ with almost 1 mSTB/d. The difference is due to the linear and cubic spline interpolation of the pressure drop curves since the pressure drop curves have a positive curvature (convex-like curves). This curvature is captured by the cubic

Table 8: Results for Case 1.

Case, Solver	Iterations (#)	Time (sec)	z^* (mSTB/d)
Case 1.1 (linear splines)			
BONMIN	0	2	77.483
CENSO	9	56	77.483
Case 1.2 (cubic splines)			
BONMIN	0	5	78.381
CENSO	17	191	78.381

spline, but is over-estimated by the linear spline (piecewise linear) interpolation. Consequently, the higher pressure drop causes a lower production for a fixed separator pressure. The two optimal solutions do however give the same optimal valve settings.

The 11 active constraints at the optimal solution are listed in Table B.15. The case has 10 wells, and one additional source node for gas lift, giving 11 DOF (when all binary variables are fixed). Hence, there are 11 active constraints at the optimal solution (in addition to 10 fixed binary variables).

At the optimal solution Well 4-7 are offline. Well 6 and 7 are set offline. Well 4 and 5 have a WCT above 50% and it is not unexpected that they are offline at the optimal solution. All online wells operate at the minimum choke differential pressure, meaning that the system is pressure constrained – the gas capacity constraint is not active and maximum gas lift is used.

To investigate the approximation error of problem **P** to the GAP model we insert the optimal valve settings into GAP and record the pressures and rates it predicts. The relative errors between the variables in GAP and Case 1.2 (cubic splines) are reported in Table 9. Most of the errors are below 1%, which is satisfactory. We do observe some propagation of error along the flowlines, and for the riser the pressure loss error is almost 4%. This may be improved upon by sampling the flowline pressure drop more densely, and accepting a higher computation time.

7.2. Case 2 and 3: Production optimization of BP subsea production system 2

In these cases we consider the production system drawn in Figure 11. The system has 13 wells, 5 flowlines, and 2 risers. Four of the wells can be routed to either of the risers. The two risers are named as follows: edge $(48, 51)$ is the east (E) riser and edge $(50, 51)$ is the west (W) riser. We refer to the flow path $44 \rightarrow 46 \rightarrow 48 \rightarrow 51$ as the E loop and the flow path $45 \rightarrow 47 \rightarrow 49 \rightarrow 50 \rightarrow 51$ as the W loop.

For brevity we assign numbers to the wells so that well i represents the well with bottomhole node index i, although the well consists of several edges and nodes. The wells are modelled using a nonlinear IPR (a piecewise function composed of a straight line and Vogel's equation), with a fixed GOR and WCT. The well parameters are listed in Table 10. As in the previous case, we model the system with three fluid phases, i.e. $\mathbf{S} = \{\text{oil}, \text{gas}, \text{wat}\}$.

The network has a total of 17 discrete edges for routing and shutting in wells. To mimic the current field operation wells 9, 10, and 13 are set offline. This leaves $2^{14} = 16,384$ routing and well status combinations. By considering the manifold routing constraints in Eq. (10) we find that the number of feasible combinations is $2^6 \cdot 3^4 = 5,184$.

Table 9: Validation using GAP of solution from CENSO on Case 1.2.

	Error (%)		
Edge e	$q_{e,\text{oil}}$	Δp_e	p_j
Wells			
(1,11)	0.26	0.18	0.14
(2,12)	0.23	0.13	0.11
(3,13)	0.21	0.10	0.09
(4,14)*	0	0	0
(5,15)*	0	0	0
(6,16)*	0	0	0
(7,17)*	0	0	0
(8,18)	0.91	0.42	0.40
(9,19)	0.47	0.38	0.37
(10,20)	1.18	0.44	0.42
Flowlines			
(21,22)	0.24	0.62	0.05
(22,23)	0.23	2.70	0.28
(23,24)	0.23	1.35	0.39
(24,26)	0.55	3.84	0.55
(26,27)	0.55	0.81	0**

* Edge is closed (well is offline) at the optimal solution.
** The separator pressure is fixed.

In addition to the common constraints described in Section 4, this case has two special constraints. Each riser has a maximum mix velocity constraint (the mix velocity is the sum of the in-situ liquid and gas velocity). These constraints limit the erosion of the risers' inner tube coating due to high velocity sand particles. The mix velocity of riser $e \in \{(48,51),(50,51)\}$ is modelled as $v_e(\mathbf{q}_e, p_j, t_e + \Delta t_e) \leq v_e^U$, where v_e^U is the upper velocity limit. Notice that the downstream pressure p_j and temperature $t_e + \Delta t_e$ are used since the velocity is calculated at the outlet, where it invariably attains its maximum value. To accurately express these important constraints, temperature and enthalpy variables are included in the formulation.

To benchmark how the model complexity added by the discrete decisions and temperature variables affect the computation time, we solve Case 2 and 3 with the various configurations described in Table 11. Note that in Case 2.1 and Case 3.1 the status and routing of the wells are set to the state of the current field operation.

7.2.1. Case 2: Optimization without energy balances

In Case 2 problem $\mathbf{P}$ is solved without temperature and enthalpy variables ($\mathbf{t}, \mathbf{\Delta t}, \mathbf{h}, \mathbf{\Delta h}$), and without the energy conservation constraints in Eqs. (6), (7), and (8). Since the riser mix velocity constraints cannot be modelled without temperature variables, they are also excluded from the problem. The nonlinear relations for pressure drop and boundary conditions are modelled with cubic splines.

The results of Case 2 are reported in Table 12. Several interesting observations can be made

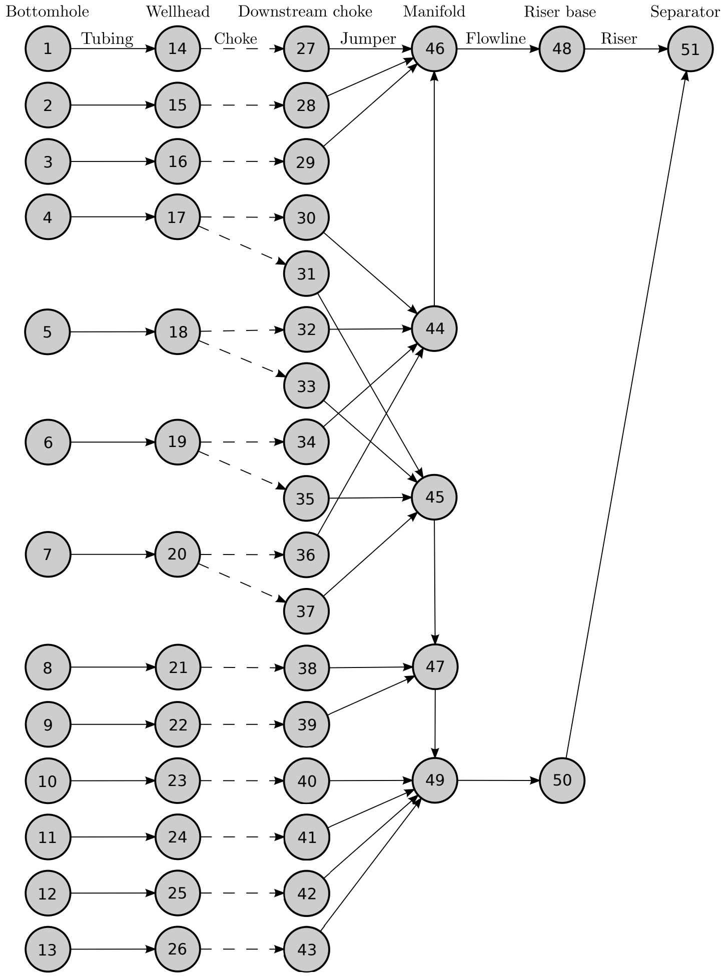

Figure 11: Topology of BP subsea production system 2.

from the results. First of all, it is clear that allowing wells to be shut in or re-routed may only increase the optimal value. The optimal value of Case 2.2 is therefore higher than Case 2.1, but lower than that of Case 2.3. Next, the local solvers seem to find good optimal solutions. In fact, in all cases except Case 2.1, all solvers are able to locate the global optimum. This may be attributed to the problem formulation in **P**, and the smoothness of the cubic splines. Finally, the exponential increase in computation time becomes distinct when globally solving Case 2.3 with CENSO.

In Case 2.3, CENSO finds the solution $z^* = 143.875$ mSTB/d. This solution is verified by running the GAP model with the optimal valve settings. This gives a production of 143.936 mSTB/d; a relative difference of 0.04%. Compared to the optimal solution from the proprietary solver in Case 2.1, the increase in production is 0.56%.

The active constraints at the optimal solution of Case 2.3 are listed in Table B.16. As indicated in the table, Well 3 is offline in the optimal solution. This is not surprising since it is a

Table 10: Well parameters for Case 2 and 3.

Well i	PI (rank #)	GOR (scf/STB)	WCT (%)	p_i^L (bara)
1	10	2400	45	280
2	5	900	0	190
3	8	4500	30	220
4	2	1500	65	0
5	3	1800	35	220
6	11	3300	0	210
7	6	2900	5	220
8	12	4200	0	220
9*	1	0	0	0
10*	7	0	0	0
11	9	900	20	200
12	4	900	0	200
13*	13	0	0	0

* Well is offline. ** Values are rounded for commercial reasons.

Table 11: Configurations in Case 2 and 3.

Case	Well status	Well routing	Energy balances	Riser velocity constraints
Case 2.1	–	–	–	–
Case 2.2	✓	–	–	–
Case 2.3	✓	✓	–	–
Case 3.1	–	–	✓	✓
Case 3.2	✓	–	✓	✓
Case 3.3	✓	✓	✓	✓

Yes: ✓, No: –

weak producer and the well with the highest GOR. Well 8, having the second highest GOR, is choked to hit the gas capacity constraint on the total gas production. The rest of the wells operate at maximum capacity, i.e. at their draw-down pressure or minimum choke differential pressure.

7.2.2. Case 3: Optimization with energy balances

The full Problem **P** is solved with temperature and enthalpy variables. The previously described riser mix velocity constraints are included in the problem formulation to guard against solutions susceptible to high erosion rates.

Note that Case 3 includes all constraints of Case 2, in addition to the energy balances and riser mix velocity constraints. Thus, the optimal values in Case 2 are necessarily lower bounds on the optimal values in Case 3.

The results for Case 3 are presented in Table 13. In Case 3.1 IPOPT, BONMIN, and CENSO finds the same (globally) optimal solution. The same solution is found by BONMIN and CENSO in Case 3.2. As reported, CENSO requires more than 1 hour to find and certify a global optimum.

Table 12: Results for Case 2.

Case, Solver	Iterations (#)	Time (sec)	z^* (mSTB/d)
Case 2.1			
Proprietary solver	–	9000	143.139
IPOPT	37	0.1	143.435
BONMIN	0*	5	143.435
CENSO	30	280	143.435
Case 2.2			
BONMIN	761	44	143.763
CENSO	19	314	143.763
Case 2.3			
BONMIN	4386	146	143.875
CENSO	89	1870	143.875

* Problem has no integer variables and is solved at the root node.

In Case 3.3, BONMIN fails to locate the same or a better optimum than Case 3.2 and 3.1. This happens because BONMIN mistakenly cuts away the optimum during its search. By comparing Case 3.1 and 3.2 we notice that, as in Case 2, the option to turn off wells does not seem to have a large impact on the solution time.

CENSO finds the same optimal solution in all three cases, with the active constraints listed in Table B.17. As indicated, Well 3 and 8 act as "swing producers" and are adjusted to hit the E and W riser mix velocity constraint, respectively. Wells 9, 10, and 13 are shut in. The rest of the wells operate at maximum capacity, i.e. in this particular case at their draw-down pressure or minimum choke differential pressure. The results indicate that shutting in or re-routing wells does not increase oil production since it is limited by the riser velocity constraints.

The optimal solution found by CENSO on Case 3.2 is 140.674 mSTB/d. This solution is verified by implementing the optimal valve settings in GAP, to give 140.650 mSTB/d. The relative difference between these two solutions is -0.02%. It is not possible to assert the accuracy of which **P** approximates the GAP model based on a single point; however, this may indicate that the accuracy increases when temperatures are included in **P**.

Prior to solving Case 3.1, the best known solution was 136.400 mSTB/d, found by the proprietary solver. The solution located by IPOPT, BONMIN, and CENSO gives a production of 140.650 mSTB/d in GAP. The potential increase in production is 4.25 mSTB/d, or 3.12%.

7.3. Pre-computations: building B-splines

Before solving **P**, B-spline approximations must be built from the samples taken from the nonlinear relations. In Table 14 we report the build times for various B-splines; the build time of a B-spline is the time it takes to solve the linear system in Eq. (20). The examples include the inflow curves, pressure drop relations, and temperature drop relations used in the cases presented previously. It is worth noting that a B-spline must be rebuilt only when the relation it approximates changes.

Table 13: Results for Case 3.

Case, Solver	Iterations (#)	Time (sec)	z^* (mSTB/d)
Case 3.1			
Proprietary solver	–	9000	136.400
IPOPT	124	0.6	140.674
BONMIN	0*	32	140.674
CENSO	59	2630	140.674
Case 3.2			
BONMIN	2432	387	140.674
CENSO	11	3670	140.674
Case 3.3			
BONMIN	16152	2328	140.462
CENSO	249	9000**	140.674

* Problem has no integer variables and is solved at the root node.

** CENSO was terminated after 9000 seconds with an optimality gap of 25.163 mSTB/d.

Table 14: B-spline build times.

Samples (#)	Dimension (#)	Degree (#)	Time (sec)
24	1	1	96×10^{-6}
24	1	3	127×10^{-6}
3773	4	1	0.2
3773	4	3	21.7
9800	4	1	3.0
9800	4	3	590.0
5184	5	1	1.4
5184	5	3	93.3

8. Concluding remarks

A framework for production optimization of multiphase flow networks has been presented. By modelling the network with a graph and the nonlinear relations in the network with B-splines, a fast solution method based on the spline-compatible MINLP solver in CENSO was devised. The solution method can solve problems formulated in the framework to global optimality. To accelerate solution times, CENSO was augmented with a primal heuristic (BONMIN) and an optimality-based bounds tightening technique from the literature. Together with a DOF analysis, this allowed us to reduce the number of sBB tree branches considerably.

In addition to the theoretical contributions outlined above, we have performed a benchmark study where the solution method is compared to several other nonlinear programming methods. The study involves three realistic cases defined using two subsea production system models provided by BP. The findings are summarized below.

- The formulation in **P** proved flexible and allowed us to model the cases in the benchmark study. The formulation includes nonlinear energy balances to model the transportation of energy with higher detail than previous works on petroleum production optimization. Features such as daisy-chained manifolds, lift gas injection, routing, and velocity constraints were easy to include.

- The B-spline surrogate models were sufficiently accurate to be used in production optimization of real cases. Inflow performance curves, pressure and temperature drop correlations, and velocity maps were successfully modelled with splines.

- CENSO was able to successfully solve several realistic cases to global optimality. However, as the solution time increases exponentially with the size of the problem (number of complicating variables), we found that the global solution method was not viable for daily production optimization of the largest case, namely Case 3.3.

- The local solvers IPOPT and BONMIN were able to successfully solve problems formulated with **P** to local optimality. In all cases except one, they located the global optimum certified by CENSO. This leads us to believe that the NLP relaxation of **P** is near convex in large portions of the feasible region. We attribute the consistency of the results to the smoothness and derivatives of the cubic B-splines, and to the linear participation of the integer variables in **P**.

- The local and global solvers are complementary in the sense that the local solvers provide fast results for complex problems and are thus suitable for daily production optimization. CENSO, however, can be used to certify local solutions from time to time, and also globally explore new production settings.

- In Case 3, the new methodology identified a potential increase in production of 4250 standard barrels of oil per day, or 3.12% more than the best, previously known solution. This solution was verified in the GAP simulator.

We believe that the above findings illustrate what any proficient practitioner of mathematical programming knows; that a “good” problem formulation is a requirement for fast solution times and consistency across solvers.

The speed of the new method would allow for parameter sensitivity analysis and stochastic optimization to include uncertainty in crucial model parameters (at least for small to moderately sized problems). With such approaches it would be possible to generate not only an optimal point, but an optimal operational plan for the user. This is an important step towards better decision support systems.

Acknowledgements

This work was supported by the Center for Integrated Operations in the Petroleum Industry, Trondheim, Norway.

The authors would like to thank Esmaeil Jahanshahi for his help in testing the thermodynamic model used in this paper.

Appendix A. Degree-of-freedom analysis

Here we give a degree-of-freedom (DOF) analysis for problem **P**. We denote the DOF with D, and calculate it as $D = D_c + D_d$, where D_c and D_d is the number of free continuous and binary variables, respectively. It is straightforward to verify that $D_d = |\mathbf{E}^{\mathrm{d}}|$ so we focus on calculating D_c. We perform the analysis in two steps: first, we calculate the DOF for a network without discrete edges, i.e. with $\mathbf{E}^{\mathrm{d}} = \emptyset$; second, we calculate the DOF for discrete edges.

First, we consider a network without discrete edges. According to the requirements all nodes, except sink nodes, must have exactly one leaving edge. Consequently, for a network without discrete edges the following must be true: $|\mathbf{E}| = |\mathbf{N}| - |\mathbf{N}^{\mathrm{snk}}|$. This relation between the number of edges and nodes is useful when we next attempt to eliminate variables with equality constraints.

To calculate D_c we first count the number of continuous variables to

$$\overbrace{|\mathbf{S}| \cdot |\mathbf{E}|}^{\mathbf{q}} + \overbrace{|\mathbf{N}|}^{\mathbf{p}} + \overbrace{|\mathbf{E}|}^{\mathbf{\Delta p}} + \overbrace{|\mathbf{E}|}^{\mathbf{t}} + \overbrace{|\mathbf{E}|}^{\mathbf{\Delta t}} + \overbrace{|\mathbf{E}|}^{\mathbf{h}} + \overbrace{|\mathbf{E}|}^{\mathbf{\Delta h}} . \tag{A.1}$$

A quick glance at Table 3 verifies these numbers. Remark that when we now attempt to eliminate variables we must take care to count one elimination per constraint, and to only eliminate a variable that participate in the constraint.

Starting with the flow rates **q**: we count equality constraints related to flow rates to $|\mathbf{S}| \cdot |\mathbf{N}^{\mathrm{int}}| + |\mathbf{S}| \cdot |\mathbf{N}^{\mathrm{src}}|$, which is the number of mass balances in Eq. (1) plus the number of rate boundary conditions in Eq. (11), respectively. Using the relation $|\mathbf{E}| = |\mathbf{N}| - |\mathbf{N}^{\mathrm{snk}}|$ we find that $|\mathbf{N}^{\mathrm{int}}| + |\mathbf{N}^{\mathrm{src}}| = |\mathbf{N}| - |\mathbf{N}^{\mathrm{snk}}| = |\mathbf{E}|$. Thus, there are $|\mathbf{S}| \cdot |\mathbf{E}|$ variables, $|\mathbf{S}| \cdot |\mathbf{E}|$ constraints, and zero DOF in the flow rates **q**.

We continue by counting $2(|\mathbf{E}| - |\mathbf{E}^{\mathrm{d}}|) = 2|\mathbf{E}|$ pressure drop constraints $\Delta p_e = f_e(\cdot)$ and $\Delta p_e = p_i - p_j$ (remembering that $\mathbf{E}^{\mathrm{d}} = \emptyset$). We also count $|\mathbf{N}^{\mathrm{snk}}|$ pressure boundary conditions. In total we get $2|\mathbf{E}| + |\mathbf{N}^{\mathrm{snk}}| = 2|\mathbf{E}| + |\mathbf{N}| - |\mathbf{E}| = |\mathbf{N}| + |\mathbf{E}|$ constraints, which is the same as the number of pressure variables. Thus, we have zero DOF in the pressure variables **p** and **Δp**.

In the same fashion we consider the $4|\mathbf{E}|$ temperature and enthalpy variables. From Eqs. (6), (7), and (8) we count $3|\mathbf{E}| + |\mathbf{N}^{\mathrm{int}}|$ constraints. We also have $|\mathbf{E}^{\mathrm{src}}| = |\mathbf{N}^{\mathrm{src}}|$ boundary constraints in Eq. (13) on the temperature variables. The total number of constraints is $3|\mathbf{E}| + |\mathbf{N}^{\mathrm{int}}| + |\mathbf{N}^{\mathrm{src}}| = 3|\mathbf{E}| + |\mathbf{N}| - |\mathbf{N}^{\mathrm{snk}}| = 4|\mathbf{E}|$. Thus, we find no DOF in the variables **t**, **Δt**, **h**, and **Δh**.

We conclude the first step of the analysis by establishing that there is no degree of freedom in **P** when $\mathbf{E}^{\mathrm{d}} = \emptyset$, i.e. $D = D_c + D_d = 0$.

In the second step of the analysis we let $\mathbf{E}^{\mathrm{d}} \neq \emptyset$, i.e. we allow discrete edges. We begin by considering a node with one leaving discrete edge. The discrete edge does not have the constraints $\Delta p_e = f_e(\cdot)$ and $\Delta p_e = p_i - p_j$. However, when $y_e = 1$, $\Delta p_e = p_i - p_j$ is recovered from the big-M constraint in Eq. (5). On the other hand, when $y_e = 0$, the flow rates are forced to zero ($\mathbf{q}_e = \mathbf{0}$) by the flow routing constraint in Eq. (9) (since we already have zero DOF in the flow rates we may use the boundary constraint $\zeta_{i,s}(\mathbf{q}_e, p_i) = 0$ to fix one pressure). In either case, one DOF remains. This DOF reflects different things for the two cases: for $y_e = 1$, Δp_e is free, but it relates the node pressures p_i and p_j, affecting the flow rate $\mathbf{q}_e$; for $y_e = 0$, Δp_e is free, but does not affect the flow rate since $\mathbf{q}_e = \mathbf{0}$ or the neighbouring pressures since $-M_e \leq p_i - p_j - \Delta p_e \leq M_e$ never become active. Note that there is a subtlety with the latter case ($y_e = 0$): since Δp_e cannot affect other variables it is not suited to be a branching variable (more importantly, Δp_e is not a complicating variable in **P**). It is better to branch on p_i, which is a complicating variable that may affect other variables. We conclude that for each discrete edge we get one DOF in the continuous variables, and in total $D_c = |\mathbf{E}^{\mathrm{d}}|$.

Finally, we consider the special case where nodes may have multiple leaving discrete edges. The only change in $\mathbf{P}$ is the addition of the inequality constraints for manifold routing in Eq. (10). These constraints do not alter the DOF.

We conclude the analysis by establishing that $D_c = |\mathbf{E}^{\mathrm{d}}|$ and $D_d = |\mathbf{E}^{\mathrm{d}}|$, giving $D = 2|\mathbf{E}^{\mathrm{d}}|$. The DOF is associated with the discrete edges $e \in \mathbf{E}^{\mathrm{d}}$ representing (choke) valves.

Appendix B. Case results

The active constraints at the optimal solution of some of the cases are reported in this appendix.

Table B.15: Active constraints at optimal solution of Case 1.

Well	Online	Active constraint	Lower bound	Upper bound	Solution
1	Yes	Choke Δp (bar)	0	–	0
2	Yes	Choke Δp (bar)	0	–	0
3	Yes	Choke Δp (bar)	0	–	0
4	No	Oil rate (mSTB/d)	0	–	0
5	No	Oil rate (mSTB/d)	0	–	0
6	No	Oil rate (mSTB/d)	0	–	0
7	No	Oil rate (mSTB/d)	0	–	0
8	Yes	Choke Δp (bar)	0	–	0
9	Yes	Choke Δp (bar)	0	–	0
10	Yes	Choke Δp (bar)	0	–	0
–	–	Lift gas (mmSTB/d)	0	20	20

References

Ahmed, T., et al., 2006. Reservoir engineering handbook. Gulf Professional Publishing.

Ahuja, R. K., Magnanti, T. L., Orlin, J. B., 1993. Network flows: theory, algorithms, and applications. Prentice hall.

Alperovits, E., Shamir, U., 1977. Design of optimal water distribution systems. Water resources research 13 (6), 885–900.

Aziz, K., Settari, A., 1979. Petroleum reservoir simulation. Vol. 476. Applied Science Publishers London.

Beggs, D. H., Brill, J. P., et al., 1973. A study of two-phase flow in inclined pipes. Journal of Petroleum technology 25 (05), 607–617.

Belotti, P., Cafieri, S., Lee, J., Liberti, L., 2010. Feasibility-based bounds tightening via fixed points. In: 4th Annual International Conference on Combinatorial Optimization and Applications, COCOA 2010. Springer Berlin Heidelberg, Kailua-Kona, HI, USA, pp. 65–76.

Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A., 2009. Branching and bounds tightening techniques for non-convex minlp. Optimization Methods & Software 24 (4-5), 597–634.

Bieker, H. P., Slupphaug, O., Johansen, T. A., et al., 2007. Real-time production optimization of oil and gas production systems: A technology survey. SPE Production & Operations 22 (04), 382–391.

Bonami, P., Biegler, L. T., Conn, A. R., Cornuéjols, G., Grossmann, I. E., Laird, C. D., Lee, J., Lodi, A., Margot, F., Sawaya, N., et al., 2008. An algorithmic framework for convex mixed integer nonlinear programs. Discrete Optimization 5 (2), 186–204.

Codas, A., Camponogara, E., 2012. Mixed-integer linear optimization for optimal lift-gas allocation with well-separator routing. European Journal of Operational Research 217 (1), 222–231.

Codas, A., Campos, S., Camponogara, E., Gunnerud, V., Sunjerga, S., 2012. Integrated production optimization of oil fields with pressure and routing constraints: The Urucu field. Computers & Chemical Engineering 46, 178–189.

Table B.16: Active constraints for Case 2.3.

Well	Online	Active constraint	Lower bound	Upper bound	Solution
E loop					
1	Yes	Draw-down pressure (bara)	283.0	–	283.0
2	Yes	Choke Δp (bar)	0.5	–	0.5
3	No	Oil rate (mSTB/d)	0.0	–	0.0
5	Yes	Choke Δp (bar)	9.5	–	9.5
6	Yes	Choke Δp (bar)	10.0	–	10.0
W loop					
4	Yes	Choke Δp (bar)	1.0	–	1.0
7	Yes	Choke Δp (bar)	5.0	–	5.0
8*	Yes	Total gas (mmscf/d)	–	300.0	300.0
9	No	Oil rate (mSTB/d)	0.0	–	0.0
10	No	Oil rate (mSTB/d)	0.0	–	0.0
11	Yes	Choke Δp (bar)	5.5	–	5.5
12	Yes	Fixed oil rate (mSTB/d)	4.4715	4.4715	4.4715
13	No	Oil rate (mSTB/d)	0.0	–	0.0

* Well 8 is adjusted to hit the gas capacity constraint.

Cordeau, J.-F., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. Transportation science 32 (4), 380–404.

Cox, M. G., 1972. The numerical evaluation of B-splines. IMA Journal of Applied Mathematics 10 (2), 134–149.

De Boor, C., 1972. On calculating with B-splines. Journal of Approximation Theory 6 (1), 50–62.

Grimstad, B., Sandnes, A., 2014. Global optimization with spline constraints: A new branch-and-bound method based on B-splines, submitted to an international journal.

Grimstad, B., et al., 2015a. CENSO: a framework for global optimization of nonconvex, possibly spline-constrained, MINLP problems. `http://github.com/bgrimstad/censo`, accessed: 2015-05-16.

Grimstad, B., et al., 2015b. SPLINTER: a library for multivariate function approximation. `http://github.com/bgrimstad/splinter`, accessed: 2015-05-16.

Grossmann, I. E., 2002. Review of nonlinear mixed-integer and disjunctive programming techniques. Optimization and Engineering 3 (3), 227–252.

Gunnerud, V., Foss, B., 2010. Oil production optimizationa piecewise linear model, solved with two decomposition strategies. Computers & Chemical Engineering 34 (11), 1803–1812.

Gurobi Optimization, Inc., 2014. Gurobi Optimizer Reference Manual. `http://www.gurobi.com`, accessed: 2014-10-01.

Kosmidis, V. D., Perkins, J. D., Pistikopoulos, E. N., 2004. Optimization of well oil rate allocations in petroleum fields. Industrial & Engineering Chemistry Research 43 (14), 3513–3527.

Kosmidis, V. D., Perkins, J. D., Pistikopoulos, E. N., 2005. A mixed integer optimization formulation for the well scheduling problem on petroleum fields. Computers & Chemical Engineering 29 (7), 1523–1541.

Luathep, P., Sumalee, A., Lam, W. H., Li, Z.-C., Lo, H. K., 2011. Global optimization method for mixed transportation network design problem: a mixed-integer linear programming approach. Transportation Research Part B: Methodological 45 (5), 808–827.

Mandl, C. E., 1980. Evaluation and optimization of urban public transportation networks. European Journal of Operational Research 5 (6), 396–404.

Martin, A., Möller, M., Moritz, S., 2006. Mixed integer models for the stationary case of gas network optimization. Mathematical programming 105 (2-3), 563–582.

Messine, F., 2004. Deterministic global optimization using interval constraint propagation techniques. RAIRO-Operations Research 38 (04), 277–293.

Table B.17: Active constraints for Case 3.2.

Well	Online	Active constraint	Lower bound	Upper bound	Solution
E loop					
1	Yes	Draw-down pressure (bara)	283.0	–	283.0
2	Yes	Choke Δp (bar)	0.5	–	0.5
3	Yes	Mixed velocity, E riser (m/s)*	–	v_e^U	v_e^U
5	Yes	Choke Δp (bar)	9.5	–	9.5
7	Yes	Choke Δp (bar)	5.0	–	5.0
W loop					
4	Yes	Choke Δp (bar)	1.0	–	1.0
6	Yes	Choke Δp (bar)	10.0	–	10.0
8	Yes	Mixed velocity, W riser (m/s)*	–	v_e^U	v_e^U
9	No	Oil rate (mSTB/d)	0.0	–	0.0
10	No	Oil rate (mSTB/d)	0.0	–	0.0
11	Yes	Choke Δp (bar)	5.5	–	5.5
12	Yes	Fixed oil rate (mSTB/d)	4.4715	4.4715	4.4715
13	No	Oil rate (mSTB/d)	0.0	–	0.0

* Velocities not displayed for commercial reasons.

Misener, R., Floudas, C., 2010. Piecewise-linear approximations of multidimensional functions. Journal of optimization theory and applications 145 (1), 120–147.

Misener, R., Gounaris, C. E., Floudas, C. A., 2009. Global optimization of gas lifting operations: A comparative study of piecewise linear formulations. Industrial & Engineering Chemistry Research 48 (13), 6098–6104.

Mjaavatten, A., Aasheim, R., Saelid, S., Groenning, O., et al., 2008. A model for gas coning and rate-dependent gas/oil ratio in an oil-rim resevoir. SPE Reservoir Evaluation & Engineering 11 (05), 842–847.

Natali, J. a. M., Pinto, J. M., 2009. Piecewise polynomial interpolations and approximations of one-dimensional functions through mixed integer linear programming. Optimization Methods and Software 24 (4-5), 783–803.

Petroleum Experts Ltd., 2014. Integrated Production Modelling software (IPM). `http://www.petex.com/products/`, accessed: 2014-10-29.

Piegl, L. A., Tiller, W., 1997. The NURBS book. Springer.

Raghunathan, A. U., 2013. Global optimization of nonlinear network design. SIAM Journal on Optimization 23 (1), 268–295.

Rashid, K., Bailey, W., Couët, B., 2012. A survey of methods for gas-lift optimization. Modelling and Simulation in Engineering 2012, 24.

Ryoo, H. S., Sahinidis, N. V., 1996. A branch-and-reduce approach to global optimization. Journal of Global Optimization 8 (2), 107–138.

Sahinidis, N. V., 2003. Global Optimization and Constraint Satisfaction. In: First International Workshop on Global Constraint Optimization and Constraint Satisfaction, COCOS 2002. Valbonne-Sophia Antipolis, France, pp. 1–16.

Schumaker, L. L., 2007. Spline Functions: Basic Theory, 3rd Edition. Cambridge University Press.

Silva, T. L., Camponogara, E., 2014. A computational analysis of multidimensional piecewise-linear models with applications to oil production optimization. European Journal of Operational Research 232 (3), 630–642.

Stenhouse, B., Woodman, M., Griffiths, P., 2010. Model Based Operational Support - Adding Assurance to Operational Decision Making. In: SPE Intelligent Energy. Utrecht, The Netherlands.

Teixeira, A. F., Barreto, F. P., Rosa, V. R., Arraes, F. F., Stender, A. S., 2013. Model Based Production Optimization Applied to Offshore Fields. In: Offshore Technology Conference. Rio de Janeiro, Brazil.

Vielma, J. P., Ahmed, S., Nemhauser, G., 2010. Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. Operations research 58 (2), 303–315.

Vielma, J. P., Nemhauser, G. L., 2011. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. Mathematical Programming 128 (1-2), 49–72.

Wächter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming 106 (1), 25–57.

Wang, P., 2003. Development and applications of production optimization techniques for petroleum fields. Ph.D. thesis, Stanford University.

Zamora, J. M., Grossmann, I. E., 1999. A branch and contract algorithm for problems with concave univariate, bilinear and linear fractional terms. Journal of Global Optimization 14 (3), 217–249.