Optim Eng (2017) 18:133–154
DOI 10.1007/s11081-016-9312-7

Unit-operation nonlinear modeling for planning and scheduling applications

Jeffrey D. Kelly[1] · Danielle Zyngier[2]

Received: 24 April 2015 / Revised: 22 December 2015 / Accepted: 2 January 2016 /
Published online: 1 February 2016

Abstract The focus of this paper is to detail the quantity and quality modeling aspects of production flowsheets found in all process industries. Production flowsheets are typically at a higher-level than process flowsheets given that in many cases more direct business or economic related decisions are being made such as maximizing profit and performance for the overall plant and/or for several integrated plants together with shared resources. These decisions are usually planning and scheduling related, often referred to as production control, which require a larger spatial and temporal scope compared to more myopic process flowsheets which detail the steady or unsteady-state material, energy and momentum balances of a particular process unit-operation over a relatively short time horizon. This implies that simpler but still representative mathematical models of the individual processes are necessary in order to solve the multi time-period nonlinear system using nonlinear optimizers such as successive linear programming and sequential quadratic programming. In this paper we describe six types of unit-operation models which can be used as fundamental building blocks or objects to formulate large production flowsheets. In addition, we articulate the differences between continuous and batch processes while also discussing several other important implementation issues regarding the use of these unit-operation models within a decision-making system. It is useful to also note that the quantity and quality modeling system described in this paper complements the quantity and logic modeling used to describe production and inventory systems outlined in Zyngier and Kelly

✉ Jeffrey D. Kelly
jdkelly@industrialgorithms.ca

Danielle Zyngier
dzyngier@hatch.ca

[1] Industrial Algorithms, 15 St. Andrews Road, Toronto, ON M1P 4C3, Canada

[2] Hatch, 4342 Queen Street, Niagara Falls L2E 7J7, Canada

(Optimization and logistics challenges in the enterprise, Springer, New York 61–95, 2009).

Keywords Continuous and batch processes · Convergent and divergent flow processes · Blenders · Splitters · Separators · Reactors · Fractionators · Black-boxes

1 Introduction

In typical process industries such as oil and gas processing, petroleum refining, petrochemicals, bulk and specialty chemicals, food and beverage, pharmaceuticals, polymers and plastics, value-add is achieved through the batch and continuous unit-operations of blending, reacting (transforming) and separating. There are also other types of hybrid processes found in the process industries such as metals smelting and casting, pulp and paper and meat processing where there are batch and dimensional (geometry or shape) unit-operations with usually pipe-less transfers of materials. On the other hand in discrete industries (apparel, appliances, automotive, electronics, instruments, furniture, etc.), value is added through activities, tasks or phases such as machining, assembling and trimming. An important differentiator between the two industries is the breakdown of production into three key dimensions: quantity, logic and quality (QLQ). The discrete industries, in spite of having the dimension of quality, deal mainly with the aspects of quantity and logic when managing their production. In this case, quantity relates to the amount or number of items whereas logic represents the operating rules or manufacturing procedural details. Quality in the context of this paper refers to the components (compositions or concentrations), properties and conditions of the material or stock being processed by the different types of unit-operations. Quality is also defined as the amount of variability in the performance of the company and in the traits, attributes or characteristics of the company's products (Bodington 1995) where it is this definition that ties the aspects of quality across both the discrete and process industries.

With respect to the tractability of the models, considering all three types of QLQ variables and constraints simultaneously becomes both discrete and nonlinear due to the binary and integer nature of the logic variables and the multilinear terms of quantity times quality. In order to avoid solving a mixed-integer nonlinear problem, which is still challenging for industrial-sized problems, these models can be decomposed or separated into two simpler sub-models, namely a logistics (quantity and logic) and a quality (quantity and quality) optimization. The logistics model solves for quantity and logic variables subject to quantity and logic balances and constraints (Kelly and Mann 2003; Kelly 2004a, 2006 and Zyngier and Kelly 2009). Quality optimization solves for quantity and quality variables subject to quantity and quality balances and constraints (Kelly 2004a, c) usually after the logic variables have been fixed at the values obtained from the solution of the logistics optimization. Fortunately solving for logistics first then quality is actually somewhat intuitive and natural given that if the quantity and logic details are not feasible (or consistent) then the quantity and quality details will also not be feasible. Hence, this provides some level of intelligent problem solving structure by allowing for a staged or progressive workflow when making production or manufacturing decisions.

The purpose of this article is to describe how the complexities of the quality optimization problem can be formulated using several standard and straightforward process unit-operation models, which when created and connected into to a larger material flow path, forms a production flowsheet, network or superstructure of the overall system being optimized or solved. This type of production flowsheeting is overviewed in Kelly (2004b) and can also handle the nuances of having several modes of operation attached to the same physical unit or equipment. It is important to state explicitly that the process models described here are not necessarily the same type of process models found in commercial process simulators or optimizers. Our basic notion is that many types of process industry production flowsheets can be modeled adequately and accurately using several rudimentary constructs which mainly involve simplified formulations of mass-transfer and reaction-kinetic like operations (Hines and Maddox 1985; Taylor and Krishna 1993; Henley and Seader 1981; Levenspiel 1998, Smith et al. 2000) but modeled in a relatively simple fashion. Although energy-transfer and momentum-transfer as well as hydraulic operations are also associated with rigorous mass-transfer and reaction-kinetic operations, and are crucial to all process industry unit-operations, they can be easily abstracted at the level of production modeling for the purposes of planning, scheduling and even control off-line and on-line decision-making. In exceptional cases however, usually due to process bottlenecks or impingements, fundamental or first-principle laws of conservation models (Pantelides and Renfro 2011) can be included instead of the simplified model(s) where specifically required at the expensive of including more detailed physical properties. Therefore, from a production flowsheeting point of view, including a larger amount of macroscopic production unit-operation models versus microscopic process models is more desirable provided the accuracy and tractability are acceptable which can be enhanced by the inclusion of measurement feedback (parameter estimation and data reconciliation).

As previously mentioned with regard to processing, production can be categorized into the following types of operations: blending, reacting, separating, storing and transferring. The first three can be either continuous or batch (or dimensional) in nature and are given our primary attention. Generally in the process industries, materials are processed in the gas, liquid or solid phase and combinations thereof. Furthermore, we can divide blending and separating operations into either "chemical" or "mechanical" blending and separating which refers to gas and liquid versus solids processing respectively. Reacting processes mainly involve gas and liquid phase chemical reactions and are rarely found in solids processing except for the existence of a solid catalyst concerning the former where there is really no reacting equivalent in the discrete industries.

The sections to follow discuss the quantity and quality modeling of both continuous and batch processes with six basic types of unit-operation models described: blenders, splitters, separators, reactors, fractionators and black-boxes. Each of these models possesses a key or significant modeling aspect relating to how flows in and out of the unit-operation are processed and they are called: recipes, ratios, recoveries, reactions, renderings and relationships respectively. We also address other pertinent topics such as solving, integrating and updating the models, thus completing our discussion.

2 Continuous versus batch operations/steady versus unsteady-state

Batch processing is very similar to continuous processing in the sense of its blending, reacting and separating phenomena except that in a batch process at least one of the feed and/or product materials entering and exiting the process is accumulated over time inside the unit for the duration of its processing. For instance, a batch distillation operation consists of an initial quantity of feed material being batched into a piece of equipment and during its operation, one or more product materials can be continuously or semi-continuously drawn out (Henley and Seader 1981; Phismister and Seider 2001). In a continuous process none of the feeds or products remains inside the unit such that the mass of material charged equals the mass of material discharged for any time interval, window or period during its operation i.e., there is a negligible rate of accumulation. A batch process is usually characterized by its quantity-size or batch-size whereas a continuous process is described by its quantity-size per time or charge-rate. Batch processes have inventory or holdup whereas continuous processes have zero inventory or holdup.

In continuous processes, multiple processing stages appear in the spatial dimension of the process, whereas for batch processes, time can be considered a processing stage in the temporal dimension. Stated more formally, batch processes involve “inter-temporal transfer” of quantity and quality whereas continuous processes do not. Both continuous and batch processes involve “inter-spatial transfer” of quantity and quality which simply means there is material or matter exchanged from one or more feed or inlet streams to one or more product or outlet streams within the same equipment as well as exchanges between different pieces of equipment within the production flowsheet. In chemical engineering design of continuous mass-transfer processes (Henley and Seader 1981; Hines and Maddox 1985; Towler and Sinnott 2008) each processing stage usually has the feeds and products at some equilibrium or steady-state. From this, the number of equilibrium stages is determined and the actual number of stages is decided assuming some level of non-ideality. From the perspective of a continuous process, such as a super-fractionator separating a mixture of propane and propylene, the number of stages represents the trays or amount of packing found inside the vessel, which also determines the spatial height of the tower (Strigle 1987). In contrast, a batch process does not necessarily have physical stages per se: the physical size of the batch processing unit is determined by the time needed to achieve the required quantity and quality specifications in the product or outlet streams. Thus, each time increment during the unit-operation of the batch process represents a functional stage similar to a physical stage found in continuous processes and is usually referred to as a phase in batch control and operations.

Continuous processes are sometimes denoted as: static, steady, stationary or perpetual whereas batch processes are often called: dynamic, unsteady, nonstationary or intermittent. This implies that continuous processes can achieve a sustained steady-state operation and batch processes are always in a non-steady-state operation. Albeit true, all types of processes that have the ability to hold or maintain material during their operations can exhibit some level of unsteady behavior. Conversely, even the ability of a unit to hold material over time does not necessarily imply unsteady behavior. Therefore, a distinction is also needed between steady-state and unsteady-state behavior or functionality given that a continuous process can be unsteady and a batch process can be steady so the terminology used above is actually interchangeable. What this really means from the point of view of our production unit-operation modeling is that there are two types of temporal models, whether continuous or batch processes (i.e., steady and unsteady-state) are being considered. The modeling presented here is steady-state modeling only which can be applied to both continuous and batch processes. The difference between continuous and batch processing for steady-state modeling, especially around a unit-operation material balance, is simply that for a batch process the material balance is made across a set of time-periods representing the digitized or discretized batch, cycle or processing-time of the operation whereas for a continuous process it is around a single time-period. Unsteady-state modeling for continuous and batch processes, which is not the focus of this paper, must also contend with the situation of explicitly handling the dynamics and integration of both quantity and quality accumulation or holdup. A proven approach to deal with unsteady-state modeling is to include an internal recycle stream within the unit-operation for each material that has significant holdup (HYSYS 2002) which allows the dynamic nature of the time-varying quantity and quality variables to be discretized and solved using either Euler's method or more sophisticated orthogonal or spline collocation methods (Renfro et al. 1987).

For the sake of brevity, all of the unit-operation models presented below are for steady-state continuous processes, but as mentioned, they can be straightforwardly extended to batch process steady-state modeling by formulating a material balance over a range or set of time-periods as opposed to a single time-period.

3 Blending operations with recipes

As mentioned, all process industry production processes involve three fundamental processing steps of blending, reacting or transforming and separating. This section describes the well-known operation of blending with any number of blended feeds (inputs, inlets) and only one blended product (output, outlet). Although it is possible to model blending operations with multiple products on the same blender unit, this can be easily handled using multiple blenders in parallel. Figure 1 shows the anatomy of a continuous blending operation.

In Fig. 1, three streams feed a blender blending together to form one product stream. In this figure, xif_t refers to the in-flow variables at each time-period with i = {1,2,3} and yip_t refers to the individual stream property variables where p may also be a set or list of properties for multi-property blenders. The out-flow and out-property are labeled xf_t and yp_t respectively. The overall or total flow and flow-

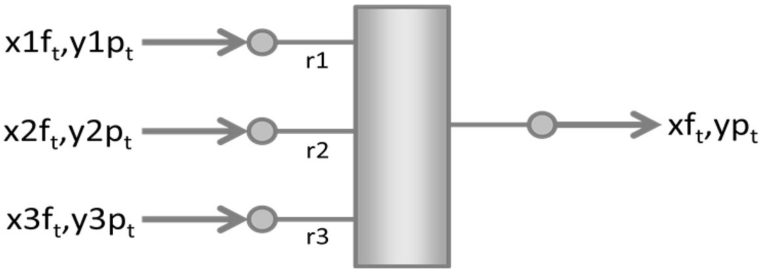

Fig. 1 A multi-*inlet*, one *outlet* and mono-property blender

property balances in open-equation or implicit-form can be seen in the following equations:

$$x1f_t + x2f_t + x3f_t - xf_t = 0,\ \forall\, t = 1\ldots NT \tag{1}$$

$$x1f_t \cdot y1p_t + x2f_t \cdot y2p_t + x3f_t \cdot y3p_t - xf_t \cdot yp_t = 0,\ \forall\, t = 1\ldots NT \tag{2}$$

where *NT* corresponds to the number of time-periods. The blend recipes r_i denote the fraction, intensity or transfer coefficient of a feed stream *i* flow with respect to or divided by the product stream flow as below:

$$xif_t - r_i \cdot xf_t = 0,\ \forall\, i = 1\ldots3 \text{ and } t = 1\ldots NT \tag{3}$$

In Eq. (3) the flow per time-period of feed stream xif_t is specified as a linear function of the product stream flow if r_i is a constant. Recipes can also be modeled as nonlinear functions of other variables where Eq. (3) would then become nonlinear.

This type of blender modeling is known as “recipe-blending” when an explicit recipe constant or variable is supplied. The other type of blend operation is called “specification-blending” which uses the recipe to only provide lower and upper bounds or inequalities on the in-flows with respect to the product out-flow. Then to specify or determine the required amount of blending, lower and upper bound specifications on the product stream property are used to drive the stream in-flow amounts. The blending model presented here can be trivially extended to perform specification-blending by including finite bounds on yp_t and replacing Eq. (3) with two lower and upper bound inequalities. In addition, this type of blending model is the same as that used in what are known as “pooling problems” found in the operations research and mathematical programming literature (Lodwick 1992; Greenberg 1995) and is also widely studied in the chemical engineering literature concerning global optimization (Tawarmalani and Sahinidis 2002; Misener and Floudas 2009). When non-ideal blend laws are required then the classic nonlinear blending of gasoline octanes by Healy et al. (1959), Morris (1975) and Rusin et al. (1981) can be employed even for non-octane properties such as distillation temperatures and vapor pressure. With these methods however, extra variables and constraints are necessary to model the synergistic and antagonistic blending effects with respect to special properties such as the amounts of saturates, olefins and aromatics as well as contaminates such as sulfur or enhancers such as lead and is beyond the scope of this paper.

4 Splitting operations with ratios

Splitting operations are essentially the dual of simple blending as shown in Fig. 2 for a mono-property where we have one in-flow and multiple out-flows. Modeling splitting operations is also very well-known and can be found in for example Quesada and Grossmann (1995).

In Fig. 2 one stream feeds a splitter splitting into three product streams. In this figure, xjf_t refers to the out-flow variables at each time-period with $j = \{1,2,3\}$ and yjp_t refers to the individual stream property variables where p may also be a set or list of properties analogous to multi-property blenders. The in-flow and in-property are labeled xf_t and yp_t respectively. The total flow balance is identical to the blender operation where all of the out-properties are simply set equal to the in-property as:

$$x1f_t + x2f_t + x3f_t - xf_t = 0, \ \forall\, t = 1\ldots NT \tag{4}$$

$$yjp_t - yp_t = 0, \ \forall\, j = 1\ldots 3 \text{ and } t = 1\ldots NT \tag{5}$$

The split ratios r_j denote the fraction of a product stream j flow with respect to or divided by the feed stream flow as below:

$$xjf_t - r_j \cdot xf_t = 0, \ \forall\, j = 1\ldots 3 \text{ and } t = 1\ldots NT \tag{6}$$

where the same modeling rules apply for split ratios as blend recipes including the notion of “ratio-splitting” and “specification-splitting”.

5 Separating operations with recoveries

Separating operations are again well-known in the chemical engineering literature given that they are ubiquitous in almost all process industries. The modeling of these types of operations can be taken from the process design and synthesis

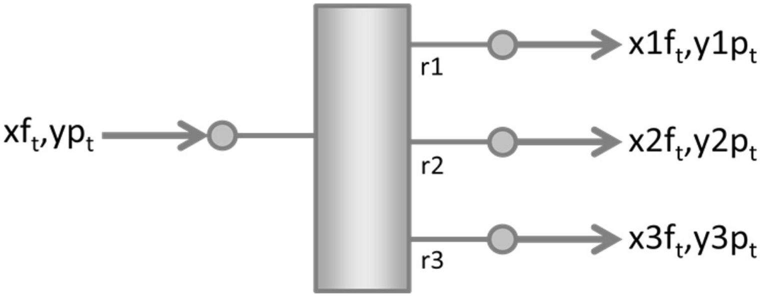

Fig. 2 A one *inlet*, multi-*outlet* and mono-property splitter

modeling found in Floudas (1995) or Towler and Sinnott (2008). Usually a separator is defined to have one feed stream and two or more product streams with two or more components and are primarily designed to be simplifications of more rigorous vapor–liquid separation processes such as equilibrium flash, distillation, fractionation and evaporation although they can also be used to model solid–solid separations of fines and lumps in mining and minerals processing (i.e., crushing beneficiation).

Figure 3 describes a one in-flow and three out-flow separator with two components. The overall flow balance is the same as both the blender and splitter i.e.,

$$x1f_t + x2f_t + x3f_t - xf_t = 0,\ \forall\, t = 1\ldots NT \tag{7}$$

And, similar to the blender for properties, there are individual bilinear flow-component balances for each component as:

$$x1f_t \cdot y1c1_t + x2f_t \cdot y2c1_t + x3f_t \cdot y3c1_t - xf_t \cdot yc1_t = 0,\ \forall\, t = 1\ldots NT \tag{8a}$$

$$x1f_t \cdot y1c2_t + x2f_t \cdot y2c2_t + x3f_t \cdot y3c2_t - xf_t \cdot yc2_t = 0,\ \forall\, t = 1\ldots NT \tag{8b}$$

In addition, for each outlet stream we also require that the sum of the components in each stream must equal unity as follows which provides the necessary stream component normalization:

$$yjc1_t + yjc2_t - 1 = 0, \forall\, j = 1\ldots 3 \quad \text{and} \quad t = 1\ldots NT \tag{9}$$

Unfortunately these equations alone are not enough to complete the modeling of these basic separation processes given that the distribution or allocation of how much of a particular component in the feed is transferred or propagated to the products has not been defined. First-principles models of rigorous separation processes would include equilibrium, mass transfer, transport phenomena, energy, etc. effects which are beyond the scope of this paper so instead it is common practice to employ what are known as component recoveries. These recoveries are similar to the blender recipes and splitter ratios but are also component dependent and are not just stream dependent. Recoveries can also be constant or determined from a nonlinear formula or function of other variables and may be specified with lower and upper bounds when component recovery ranges are required. For

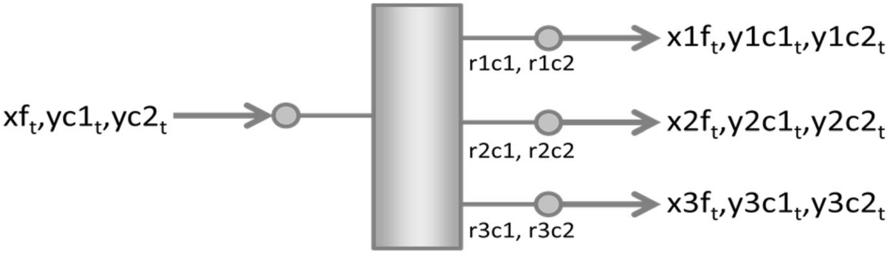

Fig. 3 A one *inlet*, multi-*outlet* and multi-component separator

instance, the recovery of the second component from the feed into the middle product is written below where $r_{j,c}$ is the recovery on the *jth* outlet stream for component *c*:

$$r_{j=2,c=2} \cdot xf_t \cdot yc2_t - x2f_t \cdot y2c2_t = 0, \ \forall \ t = 1..NT \tag{10}$$

Each component on each outlet stream would have a component recovery supplied where a common and straightforward convention of light and heavy key components would force certain recoveries to zero or one depending on the component's relative volatility compared to the key components.

6 Reacting operations with reactions

Reacting operations are somewhat unique to the process industries given that completely new species, compounds or materials are created by chemical reactions. Obviously mass is neither created nor destroyed however from the point of view of one species, say a reactant, reagent or a feed stream component, it may not appear in the product stream composition if it is completely consumed by the reaction(s). Details on the many different types of reaction structures that are encountered by chemical engineers when designing reactors can be found in Levenspiel (1998) and Worstell (2001). Concurrent (parallel), consecutive (serial) and competitive types of reaction systems with their Arrhenius rates-of-reaction as functions of frequency-factor, catalyst activity, temperature, pressure and reactant and intermediate concentrations must be properly understood before a chemical reactor can be designed and specified. However, from the perspective of planning, scheduling and control much less detailed knowledge of the reaction kinetics can be afforded. Hence, we essentially model what are known as stoichiometric or conversion reactors where the reaction kinetics may be unknown (or unimportant) but the stoichiometry and extent-of- reactions are known (Towler and Sinnott, 2008) and these types of reactor models are useful in practice (Slaback, 2004).

Consider the following two combustion reactions when burning methane gas (CH_4) in the presence of air, a mixture of oxygen (O_2) and nitrogen (N_2), producing carbon dioxide (CO_2), carbon monoxide (CO) and water vapor (H_2O):

$$\mathrm{CH_4 + 2O_2 + 0N_2 \rightarrow CO_2 + 2H_2O + 0N_2} \tag{11a}$$

$$\mathrm{CH_4 + \frac{3}{2}O_2 + 0N_2 \rightarrow CO + 2H_2O + 0N_2} \tag{11b}$$

Figure 4 shows the (stoichiometric/conversion) multi-component reactor with the six components prefixed with *y1c* and *y2c* for inlet stream 1 and outlet stream 2 respectively. The stoichiometry states the molar requirements for each of the reactions based on an extent-of-reaction variable. The extent-of-reaction is similar to a flow through the unit-operation and there is one extent-of-reaction or reaction flow for each reaction. Although the stoichiometric coefficients are usually specified

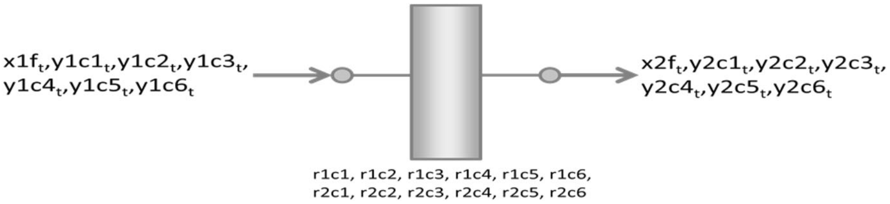
x1f$_t$,y1c1$_t$,y1c2$_t$,y1c3$_t$,
y1c4$_t$,y1c5$_t$,y1c6$_t$
x2f$_t$,y2c1$_t$,y2c2$_t$,y2c3$_t$,
y2c4$_t$,y2c5$_t$,y2c6$_t$
r1c1, r1c2, r1c3, r1c4, r1c5, r1c6,
r2c1, r2c2, r2c3, r2c4, r2c5, r2c6

Fig. 4 A one *inlet*, one *outlet*, multi-component and multi-reaction reactor

as molar, other bases such as mass or volume may be used provided the coefficients and the extent-of-reaction are both basis and dimensionally consistent.

The first set of equations are component balances for each reactant and product species which states that each component or species must conserve at the molecular or atomic level as follows:

$$x1f_t \cdot y1ck_t - x2f_t \cdot y2ck_t + \sum_{g=1}^{2} x_{g,t} \cdot r_{g,c=k} = 0, \forall\, k = 1..6 \quad \text{and} \quad t = 1..NT \tag{12}$$

where *g* is the chemical reaction index, *k* is the component index, $x_{g},_{t}$ is the extent-of-reaction and $r_{\mathrm{g,c}}$ is the component's stoichiometric coefficient involved in each reaction. The last term of course in Eq. (12) represents the amount of component that is consumed and produced by all reactions where $r_{\mathrm{g,c}}$ is negative for consumption and positive for production. There may or may not be an overall quantity balance i.e., simply flow in $x1f_t$ equals flow out $x2f_t$ given that the law of conservation may not be obeyed depending on the basis chosen. The second set of equations to complete the reactor operation is to ensure that the component fractions or compositions within the outlet stream are normalized identical to the separator operation i.e.,

$$\sum_{k=1}^{6} y2ck_t - 1 = 0, \ \forall\, t = 1..NT \tag{13}$$

7 Fractionating operations with renderings

Fractionating operations are again somewhat unique to the process industries given that they involve what are known in petroleum refining as "cuts" to essentially discretize a true-boiling-point" curve (TBP) (Gary and Handwerk, 1994) of a wide boiling-point range crude-oil mixture into smaller boiling-point mixtures of usually unknown hydrocarbon species that boil-off at certain temperatures and pressures as defined by an ASTM or API test method. There is also the well-known term "pseudo-component" (Eckert and Vanek, 2009) used in process simulators and optimizers to represent the temperature distribution of complex hydrocarbon mixtures in rigorous distillation, separation and fractionation processes involving detailed enthalpy balances and vapor–liquid equilibrium where cuts are similar to this except that their corresponding critical properties are not required. The fractionating operations considered here are referred to as basic fractionation given that we do not assume that a cut's temperature boiling-point range can be varied as part of the simulation or optimization. Advanced fractionation techniques can be found in Alattas et al. (2011) for example where the cut-point temperature can be manipulated and can be a better approach than using what is commonly known as "swing-cuts" to distinguish situations where the fraction or flow of one or more cuts that make-up two adjacent streams can be varied proportionately. In addition, a fractionator is more complex than either the splitter or separator given the existence of cuts or internal sub or meta-components but all three are divergent flow path unit-operations.

Figure 5 highlights a basic fractionator with one inlet stream, three outlet streams, a total of four components and five cuts distributed on each outlet stream accordingly. The fundamental idea of modeling a fractionating operation is to characterize each component into an arbitrary distribution or allotment of one or more cuts where this is usually referred to collectively as the component's assay-data. Then, we need to allocate or render each individual cut to an outlet stream irrespective of the component genealogy. That is, cut to outlet stream rendering is component independent i.e., we assume that there is no affect of the source component on the how the cut is distributed to a product. For the instance depicted in Fig. 5, the two components *c1* and *c2* in the one feed stream are distributed or cut into five cuts denoted as the overall cut set {*h1,h2,h3,h4,h5*} where the first stream has the cut subset {*h1,h2*}, the second stream has {*h2,h3*} and the third stream has {*h3,h4,h5*}. Notice that outlet stream 2 has an overlapping cut with outlet stream 1 (*h2*) and outlet stream 3 has an overlapping cut with outlet stream 2 (*h3*). This is known as "non-sharp" separation whereby any outlet stream can have any number of cuts assigned irrespective of the other outlet streams but usually these should be contiguous or consecutive given the nature of the boiling-point range and the there should be some conservation or continuum between how much any cut can be allocated or yielded to any stream. The two other components *c3* and *c4* on the three outlet streams are new components generated or created by the inlet components to the fractionator and can be used downstream in other cascaded or in-series

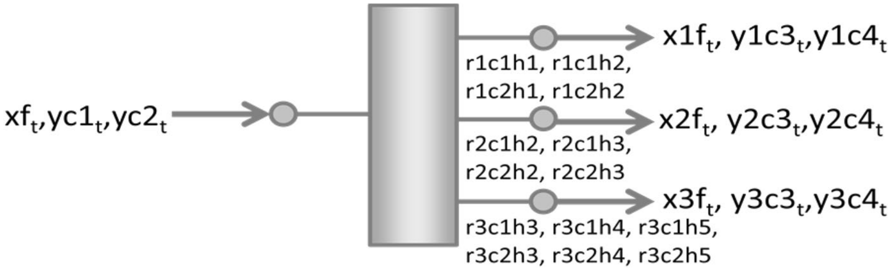

Fig. 5 A one *inlet*, multi-*outlet*, multi-component and multi-cut fractionator

fractionators if required i.e., allows for the modeling of an atmospheric tower connected downstream to one or more vacuum towers in a petroleum refinery. These new components could also be new properties generated that have specific amounts from each inlet component and from each cut, which depending on the renderings or cut-stream fractions, are pooled together on each outlet stream to form the total component or property stream value (Kelly, 2004a and Neiro and Pinto, 2004).

The above discussion provides a textual overview of the fractionator and henceforth we detail its mathematical description. The set of bilinear equations to formulate how the flow on each outlet stream is a function of the inlet flow, inlet composition and the renderings of each cut per component per outlet stream, where like the reactor, it is not required that the in-flow equals the sum of the out-flows:

$$xjf_t - xf_t \cdot \sum_{k=1}^{2} yck_t \cdot \left(\sum_{h=1}^{5} r_{j,k,h} \right) = 0, \ \forall j = 1..3 \quad \text{and} \quad t = 1..NT \tag{14}$$

where $r_{j,k,h}$ is the rendering for each outlet stream, each inlet composition and each cut respectively. The set of equations to model the composition of each new outlet component *c3* and *c4* on the three outlet streams is given by the bilinear component flow balance equation below:

$$yjckk_t \cdot xjf_t - xf_t \cdot \sum_{k=1}^{2} yck_t \cdot \left(\sum_{h=1}^{5} r_{j,k,h} \cdot wc_{k,h,kk} \right) = 0, \ \forall j = 1..3, \ kk = 3,4 \text{ and } t = 1..NT \tag{15}$$

where $wc_{k,h,kk}$ is the new component *kk* composition amount for each inlet component and cut respectively. Similarly, if properties on the outlet streams are required then we would have a different parameter $wp_{k,h,b}$ where *b* is the property in question.

8 Black-box operations with relationships

Modeling of arbitrary processes as ad hoc, bespoke, user-defined or custom input–output objects is well-known in the process system engineering domain where the previously described unit-operations may be considered as grey-box or white-box models. Especially in the process simulation and optimization fields, writing FORTRAN or other computer programming language implementations of mathematical models representing the details of the unit-operation is common-place where in simulation they are procedural (i.e., includes both closed or explicit-form modeling with embedded solving heuristics) and in optimization they are declarative (i.e., includes open or implicit-form modeling with external solving performed separately). We take a somewhat different but complementary approach to coding a custom model and that is to specify a formula or relationship of how one inlet or outlet stream variable is related to one or more other inlet and outlet stream variables as well as operating or process conditions on the unit-operation. The formula's infix or human-readable expression can then be easily lexed and parsed into either prefix or postfix byte-code (operator code and operand value), also known as reverse Polish notation (RPN), which can be evaluated at solve time as well as to provide analytical first-order derivatives numerically (Ponton, 1982).

Figure 6 shows a black-box unit-operation with three inlet streams and two outlet streams each with variable yields, one property and three process conditions denoted as $z1_t$, $z2_t$ and $z3_t$ respectively. Conditions can either be independent variables or they can be dependent on other variables in the system if needed by specifying other relationships. Yields are fundamentally identical to recipes, ratios, recoveries, reaction stoichiometry and renderings, except that when modeling a unit-operation that is both convergent and divergent, we require a characteristic charge-size if it is a continuous-unit or a batch-size if it is a batch-unit we declare as x_t. The set of equations relating a stream flow to its yield and its unit-operation's size is as follows:

$$xnf_t - yn_t \cdot x_t = 0,\ \forall\, n = 1\ldots5 \text{ and } t = 1\ldots NT \tag{16}$$

Each yield may be an arbitrary nonlinear function or formula of the conditions (or any other variable) such as:

$$yn_t - ryn(z1_t, z2_t, z3_t) = 0,\ \forall\, n = 1\ldots5 \text{ and } t = 1\ldots NT \tag{17}$$

and likewise, a stream property may also be any general relationship of the conditions such as:

$$ynp_t - rynp(z1_t, z2_t, z3_t) = 0,\ \forall\, n = 1\ldots5 \text{ and } t = 1\ldots NT \tag{18}$$

where *ryn()* and *rynp()* are the relationships relating the operating conditions to the stream yields and properties respectively for stream *n*.

Black-box operations may also be used to implement more detailed process sub-models which include physical properties relating to thermodynamics, fluid mechanics, reaction kinetics, equilibrium, etc. using the concept of a *coefficient*. Coefficients may represent enthalpies, entropies, fugacities, catalyst activities, heat

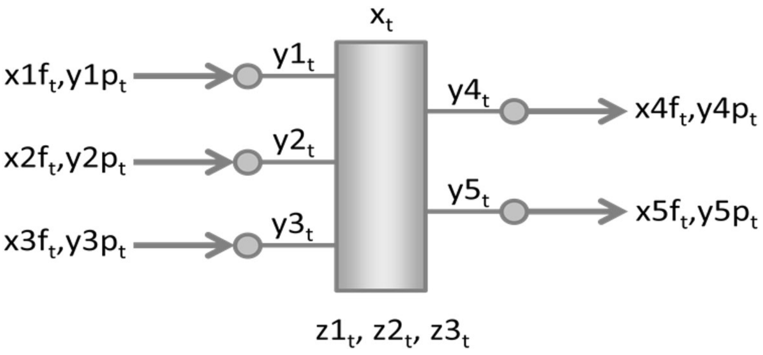

Fig. 6 A multi-*inlet*, multi-*outlet*, multi-yield, mono-property and multi-condition black-box

transfer coefficients, etc. and can be called out to a third-party physical properties package at each iteration of the nonlinear solving process. In addition, coefficients may also be fitted to actual, field or experimental data using parameter estimation and/or data reconciliation techniques on a periodic basis to calibrate the relationships.

9 Linear and nonlinear solving

As presented are moderately straightforward unit-operation building-block models to represent the complexity of a wide range of production systems found in the process industries. Modeling the system is however only a piece of the puzzle given that unless we can solve the nonlinear system of equations we cannot improve on the system's decision-making capability. Fortunately over the years there has been much improvement both in computer hardware and software as well as both company-based and now community-based linear (LP) and nonlinear programming (NLP) codes such as CLP, CONOPT, CPLEX, GLPK, GUROBI, IPOPT, LINDO, LPSOLVE, KNITRO, MOSEK, NOVA, SCIP, SOPLEX and XPRESS, just to name a few, as well as several algebraic modeling languages (AML) such as AIMMS, AMPL, GAMS, GMPL, MOSEL, MPL and ZIMPL which aid in the development and deployment of the model variables and constraints. Coding the problems in FORTRAN and C are still valid options to implement models even though providing first-order and potentially second-order partial derivatives is much more difficult than using an AML. However, with the interesting and surprisingly effective "complex-step method" (Squire and Trapp 1998), numerical derivatives of analytical quality can be computed quickly using the complex numerical algebra found in both FORTRAN and C. These derivative estimates are more accurate than finite-difference methods because the round-off errors due to the subtraction term of $f(x + h) - f(x)$ (where h is the perturbation size and $f(x)$ is the function estimate at x) is eliminated. Some nonlinear optimizers which use finite-difference methods use elaborate schemes to dynamically adjust h specific to a particular $f(x)$ and x in order to achieve some level of accuracy for the derivative but with the complex-step method perturbation sizes of 10^{-20} to 10^{-50} are not uncommon for all functions. Functions that do not have FORTRAN or C complex numerical algebra equivalents can be overloaded or modified easily using the techniques found in (Martins et al. 2003).

Another very powerful innovation is to pre-solve or pre-process the linear portion of the nonlinear system of constraints by relatively straightforward primal and dual reductions (Andersen and Andersen 1995). Both AIMMS and AMPL for example provide pre-solving techniques which are applied before the LP or NLP solvers are called to not only reduce the size of the problems by partially solving them but also to help detect and identify infeasibilities in both the model and the data. Although pre-solving has been identified by Chinneck (2008) to be somewhat useful with respect to infeasibility diagnostics, it is the authors' experiences that by only pre-solving the linear part of the nonlinear problem a large number of inconsistencies or defects can be located quickly, thus significantly increasing the ability to troubleshoot and debug large production flowsheets. Unfortunately most nonlinear programming software such as IPOPT do not recognize internally which constraints are linear or nonlinear and hence the time to diagnose issues in the optimization problem when even linear infeasibilities are present is obscured with the overall nonlinearities of the system, not to mention that the benefit of a reduced size problem cannot be gained. Although IPOPT is integrated with AMPL which can perform at the start a pre-solve before IPOPT is run but not during its iterations, there are other advantages to recognizing linear and nonlinear constraints such as in its restoration phase to handle infeasibilities i.e., infeasibilities due to the nonlinearities of the truncated linear Taylor series expansion. If a constraint is linear then no artificial or penalty variables need to be applied to restore feasibility during the solving and this is a well-known observation in SLP codes (Zhang et al. 1985).

This then leads us to how to solve these types of (linear and) nonlinear systems i.e., whether to use SLP or SQP. Again, our experiences solving many instances of these types of large-scale nonlinear models formulated using the unit-operations discussed previously is to conclude that SLP is superior especially when a commercial LP code is used as the sub-problem optimization which also implements pre-solving and scaling at every major iteration of the SLP. Dedicated planning systems for petroleum refining and petrochemical processing such as GRTMPS, MIMI, PIMS and RPMS also employ SLP (sometimes referred to as "distributed recursion") as their solver of choice which further supports our observation. One possible explanation for this, above and beyond the pre-solving and high-quality and stable linear algebra employed in commercial LP codes, is the indefiniteness of the Hessian of the Lagrange function (Renfro 2010). In production-type nonlinear models we have much fewer terms of the type "variable times itself" (i.e., squared and square-root terms and/or other irrational and transcendental functions) leading to very few diagonal elements in the Hessian compared to its nonlinear process model counterparts. Our models have more terms of the type "variable times other variables" such as a flow times a property, but few instances of flow times itself as one would find, for instance, in a hydraulic expression relating flow to head or pressure-drop. As such, an indefinite Hessian by itself or inherently leads to more vertex solutions of the nonlinear system (i.e., solutions without superbasic variables or degrees-of-freedom at the solution) where we believe and observe that SLP methods are better suited.

10 Logistics and quality optimization integration

The previous sections outlined production-type models relating to quantity and quality variables and constraints where these models are often sufficient to describe planning problems. On the other hand, for production scheduling problems, in spite of being necessary, these relationships only are not sufficient. In addition to decisions regarding quantities and qualities, scheduling problems must be able to deal with discrete decisions such as assignment and sequencing i.e., which specific unit to select and when to start an operation before or after another. In mathematical terms these decisions can be modeled through the use of logic or binary variables. A comprehensive description of these logic variables and associated modeling constraints can be found in Kelly and Zyngier (2007) and Zyngier and Kelly (2009). The combination of quantity, quality and logic variables in a single model results in a mixed-integer nonlinear programming (MINLP) model which is non-convex due to at the very least summed plus and minus bilinear terms associated with quantity times quality variables as described previously. While significant advances have been recently made in MINLP solution efficiency, a robust commercial software tool is still not available for industrial-sized problems. Hence, alternative and heuristic methods have been proposed in order to solve large-scale MINLP problems.

While some optimization approaches have been suggested such as Lagrangean decomposition (Karuppiah and Grossmann 2008), they require excessive computation for even small production scheduling problems. An alternative but heuristic approach is to successively solve the logistics (quantity and logic) MILP sub-problem first, fix the logic variables at their solution values and then solve the quality (quantity and quality) NLP sub-problem second (Kelly and Mann 2003 and Kelly 2006). After this solution is found the quality variables can be fixed and/or updated yields (recipes, ratios, recoveries, etc.) can be computed and the logistics problem re-solved. This procedure can be repeated until acceptable convergence is achieved or it is locally or globally infeasible.

This methodology resembles the well-studied approach suggested in Benders decomposition (Geoffrion 1972) where “complicating” variables (in this case, binary) are fixed such that a simpler problem may be solved which are later freed again for the new iteration. A similar method has also been applied for a different purpose, namely that of integrating decentralized decision-making systems through a hierarchical decomposition heuristic (HDH, Kelly and Zyngier 2008c). In the context of the integration between logistics and quality problems, the *coordinator* (logistics MILP sub-problem) would send what we call “logic pole-offers” to the *cooperator* (quality NLP sub-problem), which in turn would send back logic pole-offsets to the coordinator. This procedure would continue until convergence, hopefully providing at least a feasible MINLP solution.

As a final point, the integration between logistics and quality sub-problems in large-scale production scheduling problems suggested above have been implemented in a commercial product called Industrial Algorithms’ *IMPL* modeling platform (Industrial Modeling and Programming Language) using a single iteration of the procedure. Even though it is well recognized that this procedure may generate merely feasible solutions, these are both logistics and quality consistent and is therefore significantly better than other alternative commercial solutions usually employing manual trial and error simulation. It is important to highlight that hopefully sometime in the near future, the pragmatic decomposition of the MINLP into two simpler sub-problems (logistics and quality) will not be required given that MINLP solver technology is evolving. Yet, from an intelligent problem solving perspective, dissecting a large problem into smaller sub-problems (divide-and-conquer) can be a very effective way of detecting, identifying and eliminating defects in the model and data supplied.

11 Feedback and model updating

Modeling and solving alone are still not sufficient to capture the full benefit or value of enhanced decision-making using advanced model-based techniques. A continuous improvement cycle is always necessary to provide feedback from which, what we call the *plan-perform-perfect loop* (Kelly 2005), can be implemented and institutionalized to incrementally capture more and more of the value of the application that otherwise would not be realized. It is very well-known in the advanced process control field that even simple bias-updating (difference between the plant or measured and the predicted or model values) is necessary to achieve zero steady-state offset (difference between the target or setpoint and the controlled variable) of any feedback model-based regulatory controller. However as Forbes and Marlin (1994) have shown, even with bias-updating this may not be sufficient to achieve the economic performance that is expected where both the model structure and/or model parameters may be inadequately specified. Although we will never know the true model structure and parameters of any system, it behooves us to strive for that ideal.

To that end, the ability to use measured plant data from the available instrumentation system can significantly improve our model fidelity especially through off-line or on-line parameter estimation and data reconciliation. For example, by using actual and even closed-loop data from an active surge-drum level control system in a petroleum refinery, it was possible to re-fit using ordinary least-squares a simple transfer function model of the integrating process with delay to re-tune the proportional plus integral (PI) controller tuning parameters to upgrade its controller performance (Kelly 1998a). Another similar case study applied to a reactor and a tank in a small production-chain example from Kelly and Zyngier (2008a) showed that without *parameter-feedback* i.e., updating the reactor product yield parameter from past plant data, a non-zero steady-state offset between the target inventory and the tank level resulted. But, with parameter-feedback, zero steady-offset was achieved and this of course confirms the well-known observation in the control domain when applied even to simple planning and scheduling decision-making systems. This small example also points out the common misconception in post planning analysis that a detectable difference between the plan and the actual values does not unequivocally imply an execution or measurement problem. Rather that without parameter-feedback, it is impossible to discern the difference between a model inadequacy and an implementation issue.

The feedback and update model we use and propose is to first configure the production flowsheet model using the unit-operation constructs presented above with as many time-periods created as there are actual plant data sets. Then in a training or calibrating (past, present) mode, for each variable that is measured, configure another equality constraint of the form below:

$$x_t^M - g \cdot x_t - a_t = 0 \tag{19}$$

where x_t^M is the measured or actual variable value, x_t is the model variable, g is the gain which is set initially to unity and a_t is the adjustment that is minimized in a simultaneous data reconciliation and regression optimization problem (Kelly 1998b) usually with a 2-norm objective function however a 1-norm is also possible i.e., minimize the absolute values of a_t. After the data reconciliation and regression where it is also possible to fit other model parameters and after the observability, redundancy and variability estimates have been verified for uniqueness and precision (Kelly and Zyngier 2008b), it is possible for each measurement to run a simple least-squares regression to determine g from above (assuming we have several data sets). The residuals of the least-squares then become the bias estimates or *additive* uncertainty for each of the data sets and the gain g is our *multiplicative* uncertainty estimate where Eq. (19) represents an *affine* uncertainty or parity model. Then when we use the production flowsheet model in a controlling or testing (future) mode we still include Eq. (19) but now x_t^M is an optimization variable with g again fixed and the a_t variables replaced by the fixed b_t bias parameter estimates which may either be forecasts of the expected bias of the model over future time or simply fixed to the last value or some aggregated estimate from the least-squares regression. It is important to note that everywhere x_t appears in the model and it has an associated measured variable x_t^M, x_t is substituted out for or replaced by x_t^M. This is to ensure that the gain and bias updated measured variable is used in all of the future decision-making so that parameter-feedback is included where possible.

The notion of employing nonlinear dynamic data reconciliation and regression techniques, including Kalman filtering, in on-line environments is also known as *moving-horizon estimation* (MHE, Robertson et al. 1996) and is an active area in the advanced process control literature. We suggest applying the same concepts to advanced planning and scheduling decision-making whether in off-line or on-line settings except that we have slightly extended the usual bias-updating uncertainty model as shown with the use of the extra gain parameter.

12 Illustrative example

To help demonstrate the model building-blocks we present an illustrative example interconnected with a blender, reactor incorporating two reactions (R1 and R2) and two separators as well as three pools or storage unit-operations (Zyngier and Kelly 2009) as shown in Fig. 7. The diamond shapes or objects are points where materials

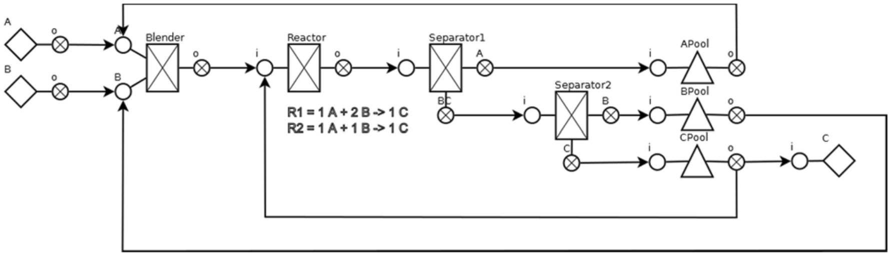

Fig. 7 An interconnected example with a blender, reactor (two reactions) and two separators

or resources can be supplied (A and B) and demanded (C). The rectangles with cross-hairs are continuous processes and the triangles represent the pools. The circles without cross-hairs are known as inlet ports and the circles with cross-hairs are outlet ports.

Every unit-operation and inlet and outlet port has three components assigned or attached for components A, B and C. All flows have a lower bound of 0 and an upper bound of 1000 and the holdup or inventory lower and upper bounds for pools APool and BPool are 0 and 0 whereas for the CPool its holdup can vary from 0 to 100. We set the blender recipes for A to vary from 0.3 to 0.4 and for B it varies from 0.6 to 0.7. The reactor has two reactions that can occur in parallel as shown in Fig. 7 where R1 requires more component B than A as specified by its reaction stoichiometry. The first separator recovers all of A on its side outlet port and all of B and C on its bottom outlet port. The flow of A recovered from the first separator flows to its pool and then from there it is sent directly back to the inlet port of A on the blender. All of the components B and C are sent to the inlet port on the second separator where all of B is recovered on its side outlet port and all of C is recovered on its bottom outlet port. Similar to A, B is sent back to its inlet port on the blender and some C is sent back to the inlet port on the reactor with a lower bound on the reactor's inlet port composition of C of 0.001 or 0.1 % to serve as an auto-catalytic component (i.e., where the reaction's product itself is used as a catalyst). Arbitrary costs of A and B are 0.1 and 0.2 respectively and the price of C is 1.0 which means that B is twice as costly as A. A single time-period into the future is used for this steady-state model where all opening inventories in the pools are specified as 0. Solving using either SLP or SQP types of nonlinear solvers results in the same objective function value of 279.72 albeit with varying numbers of iteration where due to the nature of the problem with plus and minus bilinear terms in one or more constraints, it is non-convex which can yield more than one local objective function value.

The solution only applies reaction R2 with an extent-of-reaction of 399.6 since it requires less component B (component B is twice as expensive as A) where R1's extent-of-reaction is 0. This is somewhat unrealistic in that technically both reactions should occur concurrently or simultaneously. However given that there are no reaction kinetics nor lower bounds on the extents, the optimizer is free to choose or cherry-pick a solution within the system of constraints defined. The amounts of both components A and B consumed/supplied externally are 399.6 respectively which is consistent with the stoichiometry of R2 and the amount of component C produced/demanded is of course 399.6 with no remaining holdup or inventory in the CPool at the end of the time-period. The objective function can then be back calculated as (1.0–0.1–0.2) * 399.6 = 0.7 * 399.6 = 279.72. Due to the recipe restrictions for the blender where inlet port B requires more flow than inlet port A, the recycle flow from the BPool to the blender is 199.8 and the recycle flow from the CPool to the reactor inlet port is 1.0 given the auto-catalytic requirement for at least 0.1 % of C. Ultimately, this results in the reactor's charge-size being the throughput bottleneck, active constraint or limiting step of the system i.e., 399.6 of A, 399.6 + 199.8 of B and 1.0 of C equaling the upper bound of 1000 on the reactor flow and is most likely the global optimum. All of the formulation and optimization details of this problem were performed by the *IMPL* modeling platform.

13 Conclusions

The rationale for detailing the formulation of quantity and quality variables and constraints, which exist inside all process industry production flowsheets, is to expose how to sufficiently model these types of unit-operations as key production building blocks using moderately complex formulations. Blending operations with recipes, splitting operations with ratios, separators with component recoveries, reactors with chemical reactions, fractionators with cut renderings and black-boxes with condition relationships can all be used to successfully construct large production flowsheets which are fit-for-purpose and hopefully useful. Other issues pertaining to the solving, integration and updating of these diverse models were also highlighted. Finally, this study mostly concentrated on the modeling of production flowsheets however there are many other aspects of the problem especially the issue of global optimization found in recent textbooks by Grossmann (1996) and Tawarmalani and Sahinidis (2002). Global optimization is relevant here due to the fact the models presented are non-convex and will usually exhibit local optima. Another important aspect not discussed is the compression and calibration of first-principles or rigorous models into smaller models known as meta or hybrid models (Palmer and Realff 2002). Meta and hybrid modeling technology can potentially be used to more compactly represent process models embedded inside industrial-size production flowsheets.

References

Alattas AM, Grossman IE, Palou-Rivera I (2011) Integration of nonlinear crude distillation unit models in refinery planning optimization. Ind Eng Chem Res 50:6860–6870

Andersen ED, Andersen KD (1995) Presolving in linear programming. Math Program 71:221–245

Anonymous (2002) HYSYS 3.1: dynamic modeling, Hyprotech Inc., Houston

Bodington CE (ed) (1995) Planning, scheduling and control integration in the process industries. McGraw-Hill Inc., San Francisco

Chinneck JW (2008) Feasibility and infeasibility in optimization: algorithms and computational methods. Springer Science, New York

Eckert E, Vanek T (2009) Improvements in the selection of real components forming a substitute mixture for petroleum fractions. Chem Pap 63:399–405

Floudas CA (1995) Nonlinear and mixed-integer optimization: fundamentals and applications. Oxford University Press, New York, pp 284–285

Forbes JF, Marlin TE (1994) Modeling accuracy for economic optimizing controllers: the bias update case. Ind Eng Chem Res 33:1919–1929

Gary HJ, Handwerk GE (1994) Petroleum refining technology and economics, 3rd edn. Marcel Dekker, New York

Geoffrion AM (1972) Generalized benders decomposition. J Optim Theory Appl 10(4):237–260

Greenberg HJ (1995) Analyzing the pooling problem. ORSA J Comput 7(2):205–217

Grossmann IE (ed) (1996) Global optimization in engineering design. Kluwer Academic Publishers, Boston

Healy WC, Maassen CW, Peterson RT, (1959) A new approach to blending octanes, API Refining Division, Midyear Meeting, May

Henley EJ, Seader JD (1981) Equilibrium-stage separation operations in chemical engineering. Wiley, New York

Hines AL, Maddox RN (1985) Mass transfer: fundamentals and applications. Prentice-Hall Inc., New Jersey

Karuppiah R, Grossmann IE (2008) A Lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures. J Glob Optim 41:163–186

Kelly JD (1998a) Tuning digital PI controllers for minimal variance in manipulated input moves applied to imbalanced systems with delay. Can J Chem Eng 76:967–974

Kelly JD (1998b) A regularization approach to the reconciliation of constrained data sets. Comput Chem Eng 22:1771–1788

Kelly JD (2004a) Formulating production planning models. Chem Eng Prog 100:43–50

Kelly JD (2004b) Production modeling for multimodal operations. Chem Eng Prog 100(2):44–46

Kelly JD (2004c) Formulating large-scale quantity-quality bilinear data reconciliation problems. Comput Chem Eng 28(3):357–366

Kelly JD (2005) Modeling production-chain information". Chem Eng Prog 101(2):28–31

Kelly JD (2006) Logistics: the missing link in blend scheduling optimization. Hydrocarb Process 85(6):45–51

Kelly JD, Mann JL (2003) Crude-oil blend scheduling optimization: an application with multi-million dollar benefits. Hydrocarb Process 82(6):47–53

Kelly JD, Zyngier D (2007) An improved MILP modeling of sequence-dependent switchovers for discrete-time scheduling problems. Ind Eng Chem Res 46:4964–4973

Kelly JD, Zyngier D (2008a) Continuously improve the performance of planning and scheduling models with parameter feedback. In: Proceedings of the foundations of computer-aided process operations (FOCAPO), Boston, Massachusetts, July

Kelly JD, Zyngier D (2008b) A new and improved MILP formulation to optimize observability, redundancy and precision for sensor network problems. AIChE J 54:1282–1291

Kelly JD, Zyngier D (2008c) Hierarchical decomposition heuristic for scheduling: coordinated reasoning for decentralized and distributed decision-making problems. Comput Chem Eng 32:2684–2705

Levenspiel O (1998) Chemical reaction engineering. Wiley, New York

Lodwick WA (1992) Preprocessing nonlinear functional constraints with application to the pooling problem. ORSA J Comput 4(2):119–131

Martins JRRA, Sturdza P, Alonso JJ (2003) Complex-step derivative approximation. ACM Trans Math Softw 29:245–262

Misener R, Floudas CA (2009) Advances for the pooling problem: modeling, global optimization, and computational studies. Appl Comput Math 8(1):3–22

Morris WE (1975) The interaction approach to gasoline blending, NPRA 73rd annual meeting, March, San Antonio, Texas

Neiro SMS, Pinto JM (2004) A general modeling framework for the operational planning of petroleum supply chains. Comput Chem Eng 28:871–896

Palmer K, Realff M (2002) Metamodeling approach to optimization of steady-state flowsheet simulators: model generation. Chem Eng Res Des 80:7

Pantelides CC, Renfro JG (2011) The online use of first-principles models in process operations: review, current status & future needs. In: Proceedings of the chemical process control VIII conference, Savannah, Georgia, January

Phismister JR, Seider WD (2001) Bridge the gap with semi-continuous distillation. Chem Eng Prog 97:72–78

Ponton JW (1982) The numerical evaluation of analytical derivatives. Comput Chem Eng 6:331–333

Quesada I, Grossmann IE (1995) Global optimization of bilinear process networks with multicomponent flows. Comput Chem Eng 19(2):1219–1242

Renfro, J.G., Personal communication, 2010

Renfro JG, Morshedi AM, Asbjornsen OA (1987) "Simultaneous optimization and solution of systems described by differential/algebraic equations. Comput Chem Eng 11:503–517

Robertson DG, Lee JH, Rawlings JB (1996) A moving-horizon based approach to least-squares estimation. AIChE J 42:2209–2224

Rusin MH, Chung HS, Marshall JF (1981) A transformation method for calculating the research and motor octane numbers of gasoline blends. Ind Eng Chem Fundam 20:195–204

Slaback DD (2004) A surrogate model of refinery-wide optimization. PhD dissertation, Texas Tech University

Smith JC, Harriott P, Harriot P, McCabe WL (2000) Unit operations of chemical engineering. McGraw-Hill Book Company, New York

Squire W, Trapp G (1998) Using complex variables to estimate derivatives of real functions. SIAM Rev 40:110–112

Strigle RF (1987) Random packings and packed towers: design and applications. Gulf Publishing Co., Houston

Tawarmalani M, Sahinidis NV (2002) Convexification and global optimization in continuous and mixed-integer nonlinear programming. Kluwer Academic Publishers, London

Taylor R, Krishna R (1993) Multicomponent mass transfer. Wiley, New York

Towler G, Sinnott R (2008) Chemical engineering design: principles, practice and economics of plant process design. Elesvier, London

Worstell JH (2001) Don't act like a novice about reaction engineering. Chem Eng Prog 97:68–72

Zhang J, Kim N-H, Lasdon L (1985) An improved successive linear programming algorithm. Manag Sci 31:1312–1331

Zyngier D, Kelly JD (2009) Multi-product inventory logistics modeling in the process industries. In: Wanpracha Chaovalitwongse, Kevin C. Furman, Panos M. Pardalos (Eds.) Optimization and logistics challenges in the enterprise. Springer optimization and its applications, 30, Part 1, pp 61–95. doi:10.1007/978-0-387-88617-6_2