Expanding Scope and Computational Challenges in Process Scheduling

Pedro M. Castro[a], Ignacio E. Grossmann[b,], and Qi Zhang[b,c]*

[a] Centro de Matemática Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal.

[b] Center for Advanced Process Decision-Making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

[c] Currently at BASF SE, 67063 Ludwigshafen, Germany.

Abstract

In this paper, we first present a brief overview of the area of enterprise-wide optimization and the challenges in multiscale temporal modeling as well as integration of different models for the decision levels of planning, scheduling and control. We briefly review the state-of-the-art of the scheduling area, and highlight non-conventional applications, including heat integration, pipeline scheduling and blending operations. We next address the demand side management area, a new area for scheduling of power-intensive systems, and present as a specific example, the problem of scheduling operating reserve under uncertainty for which a robust optimization approach is adopted. Finally, we address the area of integration of planning and scheduling, still a largely unsolved problem for which several approaches are reviewed, including the use of constraints of the traveling salesman problem for optimizing multiperiod planning of refineries and multisite planning and scheduling of multiproduct batch plants.

Keywords: *Planning, scheduling, demand side management, mixed-integer programming, generalized disjunctive programming.*

1. Introduction

Enterprise-wide optimization (EWO) has become a major goal in the process industries due to the increasing pressures for remaining competitive in the global marketplace. EWO involves optimizing the supply, manufacturing and distribution activities of a company to reduce costs, inventories and environmental impact,

* Corresponding author: grossmann@cmu.edu

and to maximize profits and responsiveness. Major operational items include planning, scheduling, and control. Reviews of this area can be found in Grossmann (2005; 2012).

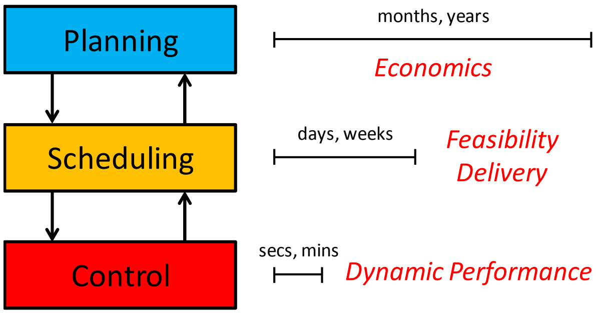


Fig. 1. Multiple times scales in planning, scheduling and control with corresponding objectives.

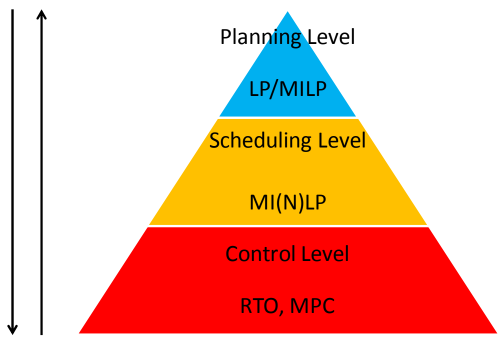


Fig. 2. Multiple optimization models at each operational level.

Major challenges in Enterprise-wide Optimization involve first the integration of multiple scales, as seen in Fig. 1 in which the time scale spans from months/years down to seconds/minutes, with objectives of each of the major functions (planning, scheduling and control) ranging from economics, feasibility to dynamic performance. A second major challenge that is a consequence of the nature of the three major decision levels is the need to coordinate different types of models as seen in Fig.2. Typically, the planning level involves LP and MILP models, the scheduling level MILP/MINLP models, and the control level MPC and RTO models. Ideally, one would like to solve and coordinate the different models as if they were solved simultaneously.

In this paper, however, we focus specifically on the development of optimization models for the scheduling level. Generally, scheduling can be defined as the problem for which we are given an existing plant with product demands over a given time horizon. The problem is to select the allocation of available resources over time in order to perform a set of tasks in order to meet the given production targets. The objectives can range from minimizing makespan to minimizing cost.

We should note that the scheduling area is relatively new in chemical engineering (it started in 1978/1979). Table 1 provides a sample of evolution of representative papers in this area, including the interface with planning and control levels.

Table 1. Representative papers on the evolution of scheduling in chemical engineering.

Reklaitis, G.V., 1978. Review of scheduling of process operations. AIChE Symp. Ser. 78, 119-133
Maudeli, A.M., Rippin, D.W.T., 1979. Production Planning and Scheduling for Multipurpose Batch Chemical Plants. Comp. Chem. Eng. 3, 199-206.
Shah, N., 1998. Single- and multisite planning and scheduling: current status and future challenges. Proceedings of FOCAPO-98, pp 75-90.
Kallrath, J., 2002. Planning and scheduling in the process industry. OR Spectrum 24, 219-250.
Floudas, C.A., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Comp. Chem. Eng. 28, 2109-2129.
Maravelias, C.T., Sung, C., 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. Comp. Chem. Eng. 33, 1919-1930.
Wassick, J., 2009. Enterprise-wide optimization in an integrated chemical complex. Comput. Chem. Eng. 33, 1950-1963
Grossmann, I.E., 2012. Advances in Mathematical Programming Models for Enterprise-Wide Optimization. Comp. Chem. Eng. 47, 2-18.
Harjunkoski, I., Maravelias, C., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014 Scope for Industrial Applications of Production Scheduling Models and Solution Methods. Comput. Chem. Eng. 62, 161-193.
Baldea, M., Harjunkoski, I., 2014. Integrated production scheduling and process control: a systematic review. Comp. Chem. Eng. 71, 377-390.
Dias, L.S., Ierapetritou, M., 2016. Integration of scheduling and control under uncertainties: review and challenges. Chem. Eng. Res. Des. 116, 98-113.

Recent extensive reviews on scheduling and their adoption in industry can be found in the papers by Mendez et al. (2006) and Harjunkoski et al (2014).

In this paper, we focus on three major topics: (a) Scheduling, (b) Demand Side Management, (c) Integration of Planning and Scheduling. For the topic on Scheduling (section 2) we present a brief review of the state-of-the-art, emphasizing the role that Generalized Disjunctive Programming has as general modeling framework for deriving optimization models for scheduling. We also address several non-conventional applications, including heat integration, pipeline scheduling and blending operations. On the topic of Demand Side Management (section 4), a new area for scheduling of power-intensive systems, we present a brief introduction, and as a specific example, the problem of scheduling operating reserve under uncertainty for which a robust optimization approach is adopted. Finally, on the topic of Integration of Planning and Scheduling (section 5) we discuss several major approaches, and focus on the use of traveling-salesman constraints for handling changeovers, and decomposition schemes such as Lagrangean decomposition and bi-level decomposition.

2. Basic Concepts

Scheduling problems and mathematical formulations for their solution can be classified according to different criteria. Below, we summarize the most commonly used, to help with the discussion of the industrial case studies presented later in the paper. In the process, we show that landmark scheduling constraints are not difficult to derive when relying on Generalized Disjunctive Programming.

2.1. Production recipe and environment

The best scheduling formulation for a problem depends on its features (Harjunkoski et al., 2014). A wide variety has been considered for problem classification but here we limit the discussion to the production recipe and the production environment, which are very much related.

A production recipe consists of a sequence of tasks involving one or more materials. Usually, it is not required to know the recipe in full detail, only its key stages. If the recipe involves intermediate states of a single material, the identity of the final product is preserved and we have a sequential facility. Sequential facilities can be divided into: single-stage, if all stages of the recipe are performed in the same equipment unit (see later Fig. 8); multi-stage, when dealing with similar production recipes for all products and each stage is linked to a specific type of unit (examples from steel and pulp plants can be found in Castro et al. 2013 and Castro et al. 2014); multipurpose, otherwise (see Fig. 3, where the units are tanks used for treating airplane parts).

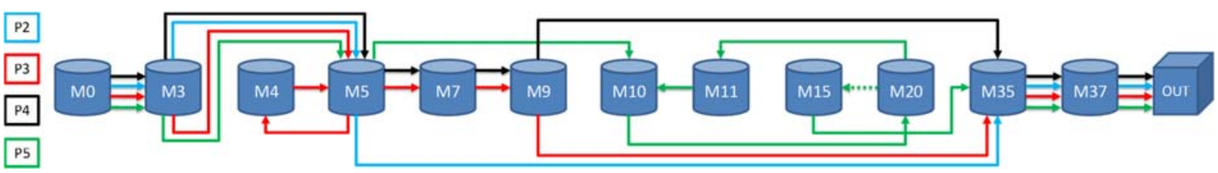


Fig. 3. Multipurpose production environment for sequential facility, also known as jobshop.

If aggregating the materials of a product recipe into one compromises the quality of the schedule, we have material-based recipes that may involve multiple products. In the famous problem by Kondili et al. (1993), represented in Fig. 6 as a Resource-Task Network (see later 2.5) for the case of batch-size and unit-dependent processing times (Ierapetritou and Floudas, 1998a; Castro et al. 2001), there are three feedstocks (Feed A-C), four intermediates (HotA, IntAB, IntBC, ImpureE) and two final products. The goal is to effectively use the two available reactors (Reac1-2) for the three different reactions (Reaction1-3), selecting batch sizes as part of the optimization and keeping inventory within limits. Other examples of network facilities can be found in the benchmark problem from the German chemical industry (Westenberger and Kallrath, 1994; Kallrath, 2002; Blomer and Gunther, 2000) and in the case study from ATOFINA Chemicals (Lin et al. 2002).

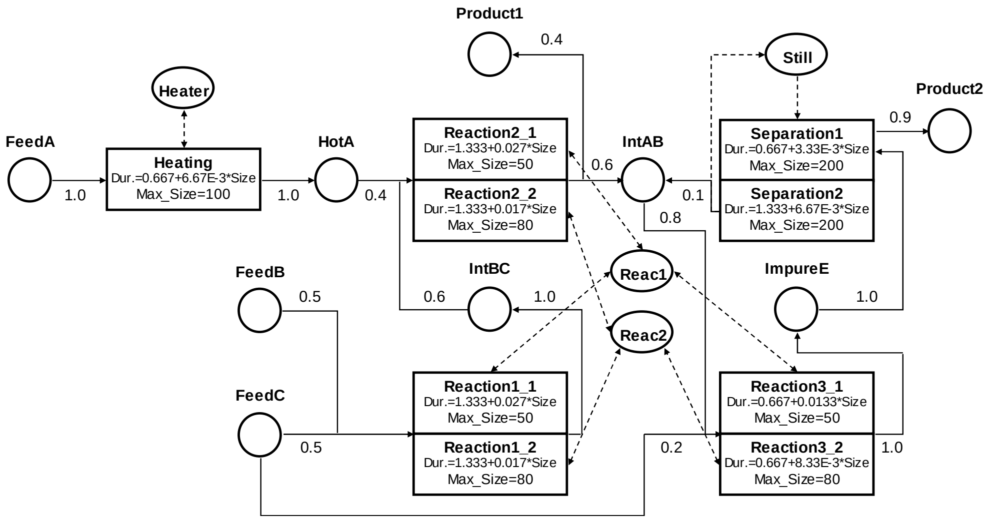


Fig. 4. Material-based production recipe for network production facility (Resource-Task Network).

2.2. Time representation

Scheduling formulations are primarily distinguished by the way they handle time. Different concepts for time representation can be found in the literature and their main purpose is to facilitate the writing of constraints that prevent tasks competing for a unary resource to occur simultaneously. They can be divided into the three different types summarized in Fig. 5.

Type A includes discrete-time representations that divide the time horizon of interest into a fixed number of time points. The most common approach is to rely on a single uniform time grid with slots of size δ (h), a parameter that should be small to consider sufficiently accurate values τ_i when converting processing times p_i (h) (typically $\tau_i = \lceil p_i/\delta \rceil$). One can also rely on multiple non-uniform time grids to reduce problem size and improve computational performance, after using an algorithm to determine the proper size for each grid (Velez and Maravelias, 2013, 2015). The number of time points is typically large and their location is known a priori, which is useful for handling external events like due dates, maintenance activities, utility availability and electricity pricing (Castro et al. 2013).

In the discrete-time grid in Fig. 5, more time points t are used than those required to identify the beginning and end of tasks. It motivated the development of grid-based continuous-time models that use fewer time points and generate smaller mathematical problems (type B). Notice that compared to our previous work in Harjunkoski et al. (2014), we no longer distinguish between single and multiple grids. The important aspect is that the number of time points needed to find the optimal solution is unknown a priori, requiring an iterative search procedure (Méndez et al., 2006). Included in this lot are the well-known single grid formulations of Schilling and Pantelides (1996), Castro et al. (2001; 2004), Maravelias and Grossmann (2003) and Sundaramoorthy and Karimi (2005); and the unit-specific formulations of Ierapetritou and Floudas (1998a,b), Shaik and Floudas (2008) and Susarla et al. (2010).

Type	Representation	Resource balances?	Time grid	Number of grids	Type of grid	Location of time points	Illustration
A	Discrete	Yes	Explicit	Single or multiple	Uniform or non-uniform	Known *a priori*	
B	Continuous	Yes	Explicit	Single or multiple	Non-uniform	Determined by optimization	
C	Continuous	No	Implicit or Explicit	Multiple	Non-uniform	Determined by optimization	

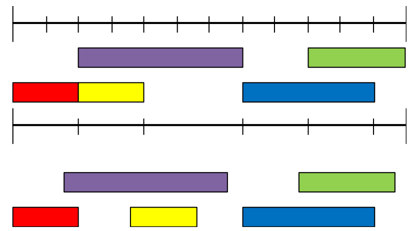


Fig. 5. Time representation concepts in scheduling formulations

Type C includes continuous-time formulations that do not rely on resource balances. They are particularly suitable for sequential production environments, where it suffices to ensure that the tasks defining the production recipe of an order take place sequentially, and that tasks of different orders competing for the same equipment do not overlap in time. Time grids are implicit when using precedence-based models, using either immediate (local) (Méndez et al., 2000; Gupta and Karimi, 2003) or general (global) precedence variables (Méndez et al., 2001; Harjunkoski and Grossmann, 2002; Méndez and Cerdá, 2003). Models with explicit time grids, identified by the presence of timing variables with a time point index (Castro and Grossmann, 2005; Castro et al., 2006; Liu and Karimi; 2007; 2008), have also been used.

Multiple time grid formulations of type C are particularly efficient when the number of time points required to find the global optimal solution is known a priori, e.g. when dealing with multistage plants with a single unit per stage. As an example, consider the real-life problem from a pulp plant with four parallel batch digesters competing for shared resources (Castro et al., 2014). To model the complex interaction during the heating stage (bottleneck), we first generated a superstructure with temperature states and used a scheduling model with resource balances to identify the optimal digester sequence, testing both discrete and continuous-time representations (single grid). More recently, we avoided the explicit definition of states by deriving the interacting timing constraints from GDP. The results in Table 2, reflect orders of magnitude differences in problem size and computational time, and highlight the importance of model selection for problem tractability. We return to this topic in section 3.3.5.

Table 2. Comparison of time representation concepts for an industrial case study.

Time representation	Discrete (single, uniform gird)	Continuous (single grid)	Continuous (multiple time grids)
Type	A	B	C
Time slots (per grid)	564	16	4
Binary variables	31583	3584	79
Total variables	60914	4435	141
Equations	29372	4435	132
Optimal solution	564	571[a]	564
Computational time (CPUs)	217	25252	0.21

[a] Suboptimal solution due to insufficient number of time points in the grid.

2.3. Generalized Disjunctive Programming (GDP)

Most scheduling problems can be formulated as mixed-integer linear programming problems. It is not uncommon to find in the literature models with over one hundred sets of constraints, some of them combining binary and continuous variables in a way that makes it difficult to understand how they work. Often, there is an abundance of big-M type of constraints that are responsible for weak linear relaxations and poor computational performance.

Generalized disjunctive programming (Balas, 1979; Raman and Grossmann, 1994) provides a high-level framework for systematic model development that is based on equations and symbolic logic. It keeps model complexity at a manageable level when dealing with systems with different types of constraints. It is particularly useful for scheduling (Castro and Grossmann, 2012), where no two problems are the same, and for which there are alternative ways of dealing with the timing element and sequencing decisions, whose relaxation in the transformed mixed-integer constraints is often difficult to predict.. GDP represents discrete/continuous logic through disjunctions and logic propositions in terms of Boolean variables. The third type of constraints uses solely continuous variables (e.g. mass balances, bounding constraints). Recent reviews on generalized disjunctive programming can be found in (Grossmann and Trespalacios, 2013; Trespalacios and Grossmann, 2014).

One of the goals of this paper is to show that well-known scheduling formulations can be formulated as the linear generalized disjunctive program in Eq. (1). Linear with respect to constraints $A_{j,k}x \leq b_{j,k}$ inside term j of disjunction k, since functions $f(x)$ and $g(x)$ of bounded nonnegative continuous variables x (superscript L/U stands for lower/upper bound) can be nonlinear, a case leading to a mixed-integer nonlinear problem (MINLP). Examples are multiperiod blending problems in petroleum refineries (Lee at al., 1996; Kolodziej et al., 2013) where nonlinearities are due to non-convex bilinear terms and binary variables appear linearly in the constraints, a special class of mixed-integer quadratically constrained problems (Castro, 2017). Boolean variables Y appear in disjunctions and logic propositions $\Omega(Y)$ and have a one-to-one correspondence with binary variables ($Y \equiv True \equiv 1$, $\neg Y \equiv False \equiv 0$). Note that we are using the exclusive OR operator $\underline{\vee}$ for separating the terms of a disjunction but inclusive ORs can also be handled (see Mostafaei and Castro, 2017).

$$\begin{aligned}
&\min f(x) \\
&\text{s.t. } g(x) \leq 0 \\
&\underline{\bigvee}_{j \in J_k} \begin{bmatrix} Y_{j,k} \\ A_{j,k}x \leq b_{j,k} \end{bmatrix} \forall k \qquad (1) \\
&\Omega(Y) = True \\
&x \in \mathbb{R}^n, 0 \leq x^L \leq x \leq x^U, Y \in \{False, True\}^m
\end{aligned}$$

2.3.1. Big-M reformulation of disjunctions

The simplest representation of the disjunctions in mixed-integer linear form is achieved by the big-M reformulation in Eq. (2). It is perhaps the reason why it is most commonly used, despite often yielding poor relaxations. The big-M vector-parameter can be made constraint (there can be multiple constraints inside a disjunctive term), term and disjunction dependent. However, modelers typically do not bother to find the tightest parameters, replacing $M_{j,k}$ with a global value (e.g. the time horizon, when dealing with timing constraints in continuous-time scheduling formulations).

$$A_{j,k}x \leq b_{j,k} + M_{j,k}(1 - Y_{j,k}) \ \forall k, j \in J_k \tag{2}$$

We should note that an improved formulation for big-M constraints for GDP problems has been reported by Trespalacios and Grossmann (2015).

2.3.2. Convex hull reformulation of disjunctions

The convex-hull reformulation (Balas, 1985) in Eq. (3), leads to MILP constraints that are at least as tight as their big-M counterparts. The disadvantage is that we need additional continuous variables, $\hat{x}_{j,k}$, and constraints. In some cases, however, it is possible to combine the resulting equations to eliminate the additional variables and constraints, leading to a sharp/compact formulation (Jeroslow and Lowe, 1984; Castro and Grossmann, 2012) with much better computational performance. Examples appearing later, include the modelling of a semi-continuous variable (Eq. 7) and timing constraints in grid-based continuous-time formulations (Eq. 19).

$$\begin{aligned}
&A_{j,k}\hat{x}_{j,k} \leq b_{j,k}Y_{j,k} \ \forall k, j \in J_k \\
&\hat{x}_{j,k}^{L}Y_{j,k} \leq \hat{x}_{j,k} \leq \hat{x}_{j,k}^{U}Y_{j,k} \ \forall k, j \in J_k \\
&x = \textstyle\sum_{j \in J_k} \hat{x}_{j,k} \ \ \forall k \\
&\sum_{j \in J_k} Y_{j,k} = 1 \ \forall k \\
&0 \leq \hat{x}_{j,k}^{L} \leq \hat{x}_{j,k} \leq \hat{x}_{j,k}^{U}
\end{aligned} \tag{3}$$

2.3.3. Reformulation of logic propositions

Constraints including solely binary variables are derived from logic propositions. The reformulation can be done by converting the logic expression into conjunctive normal form (Cloksin and Mellish, 1981; Raman and Grossmann, 1991). It consists in a conjunction of clauses connected by AND operators $\wedge$. Each clause is a disjunction connected by the inclusive OR operators $\vee$. In terms of binary variables Y, the inclusive OR is modelled by Eq. (4).

$$\textstyle\sum_{j \in J_k} Y_{j,k} \geq 1 \ \forall k \tag{4}$$

2.4. State-Task Network (STN)

The seminal paper dealing with a discrete-time formulation for scheduling batch chemical processes was due to Kondili et al. (1993) and brought us the State-Task Network representation. It is one of the most successful papers in process systems engineering with over 600 citations and #5 of all time in Computers and Chemical Engineering. The basic idea is to represent production recipes as the transformation of material states by means of tasks. A task i may have multiple input and output states s, allowing for complex interactions (e.g. heat integrated tasks, see Papageorgiou et al., 1994). A state is a node in the network that can be consumed/produced partially or entirely by different tasks, enabling the modelling of multiple processing routes, shared intermediates and recycles. One very important feature is that consumption ($\nu_{s,i,\theta} < 0$) and production ($\nu_{s,i,\theta} > 0$) can occur at intermediate points in time θ relative to the start of the task, not just at the beginning and end (a limitation of continuous-time formulations).

The mathematical formulation has three fundamental constraints: (i) material balances; (ii) capacity limitations; (iii) equipment allocation constraints. To simplify the nomenclature, we now assume that an operation that can be performed in alternative equipment units is divided into multiple tasks. Let binary variable $N_{i,t} = 1$ indicate that task i starts at time point t, nonnegative continuous variable $\xi_{i,t}$ hold the amount processed, and parameters υ_i^L and υ_i^U be their lower and upper bounds. Continuous variables $S_{s,t}$ represent the amount of state s available at t, $s_{s,t}^L$ and $s_{s,t}^U$ are their lower and upper bounds, while parameters $\pi_{s,t}$ account for external interactions (e.g. raw-material supply (>0) and product demand (<0)).

2.4.1. Material balances

The material balances in eq Eq. (5) involve solely variables of the continuous type and simply state that the net increase in the amount stored from time $t-1$ to t is the difference between the amounts produced/consumed (proportional to the amount processed) and external interactions. Notice that production of state s at the end of task i ($\theta = \tau_i$) appears at point t if the task started at point $t - \tau_i$.

$$S_{s,t} = S_{s,t-1} + \sum_i \sum_{\theta=0}^{\tau_i} \nu_{s,i,\theta}\xi_{i,t-\theta} + \pi_{s,t} \ \forall s,t \tag{5}$$

2.4.2. Capacity limitations

Capacity limitations are divided into two sets of constraints. Eq. (6) models storage limitations by enforcing upper and lower bounds, while Eq. (7) relates a task binary and continuous extent variables.

$$s_{s,t}^L \leq S_{s,t} \leq s_{s,t}^U \ \forall s,t \tag{6}$$

$$\upsilon_i^L N_{i,t} \leq \xi_{i,t} \leq \upsilon_i^U N_{i,t} \ \forall i,t \tag{7}$$

As mentioned earlier, constraints involving binary and continuous variables can systematically be derived from disjunctions. Eq. (8) enforces the material bounds if the task is executed, otherwise the amount processed is equal to zero. Eq. (7) results from the convex hull reformulation of Eq. (8).

$$\begin{bmatrix} N_{i,t} \\ \upsilon_i^L \leq \xi_{i,t} \leq \upsilon_i^U \end{bmatrix} \veebar \begin{bmatrix} \neg N_{i,t} \\ \xi_{i,t} = 0 \end{bmatrix} \forall i,t \tag{8}$$

2.4.3. Equipment allocation

If equipment j starts to perform task $i \in I_j$, then it cannot start any other task until i is finished. This was modelled by Kondili et al. (1993) using big-M type of constraints. Eq. (9) was later replaced by Eq. (10), which has a smaller domain and is tighter (Shah et al., 1993).

$$\sum_{i' \in I_j} \sum_{t'=t}^{t+\tau_i-1} N_{i',t'} - 1 \leq M(1 - N_{i,t})\ \forall j, i \in I_j, t \tag{9}$$

$$\sum_{i \in I_j} \sum_{t'=t}^{t-\tau_i+1} N_{i,t'} \leq 1\ \forall j, t \tag{10}$$

Equations limited to binary variables can be derived systematically from logic propositions. Let binary variable $Y_{i,t} = 1$ if task i is active during time slot t and $Y_t^{no\ i} = 1$ if no task is active. Clearly, at most one task can be active in unit j during slot t (Eq. 11). Being active is then equivalent to saying that the task has either started at t or at a previous slot not farther than $\tau_i - 1$ (Eq. 12). Replacing Eq. (12) into (11) and reformulating as a linear inequality, leads to Eq. (13), which is equivalent to Eq. (10).

$$\underline{\bigvee}_{i \in I_j} Y_{i,t} \underline{\bigvee} Y_t^{no\ i}\quad \forall j, t \tag{11}$$

$$Y_{i,t} \Leftrightarrow \underline{\bigvee}_{\theta=0}^{\tau_i-1} N_{i,t-\theta}\ \ \forall i, t \tag{12}$$

$$\sum_{i \in I_j} \sum_{\theta=0}^{\tau_i-1} N_{i,t'} \leq 1\ \forall j, t \tag{13}$$

2.5. Resource-Task Network (RTN)

Besides material states and equipment units, Kondili et al. (1993) also discuss how to handle utilities and manpower. In general, one can model the amount required of resource r by the combination of a constant $\mu_{r,i,\theta}$ and a variable term $\nu_{r,i,\theta}\xi_{i,t}$, paving the way for the excess resource balances of the Resource-Task Network representation (Pantelides, 1994) given in Eq. (14). Notice that structural parameters $\mu_{r,i,\theta}$ and $\nu_{r,i,\theta}$ allow modeling other cases like temporary consumption of equipment units due to task execution ($\mu_{r,i,0} = -1$, $\mu_{r,i,\tau_i} = 1$, $\nu_{r,i,\theta} = 0$), and permanent consumption/production of material resources for fixed ($\mu_{r,i,\theta} < 0$ / $\mu_{r,i,\theta} > 0$, $\nu_{r,i,\theta} = 0$) and variable batch sizes ($\mu_{r,i,\theta} = 0$, $\nu_{r,i,\theta} < 0$ / $\nu_{r,i,\theta} > 0$). Tasks with variable batch sizes can also be handled by creating subtasks, one for each discrete duration (Kondili et al., 1993; Sundaramoorthy and Maravelias, 2011), where the additional constraint for variable size and duration ensures that a single size (or size subinterval) is selected, see Fig. 6 on the right.

$$R_{r,t} = R_{r,t-1} + \sum_i \sum_{\theta=0}^{\tau_i} (\mu_{r,i,\theta} N_{i,t-\theta} + \nu_{r,i,\theta}\xi_{i,t-\theta}) + \pi_{r,t}\ \forall r, t \tag{14}$$

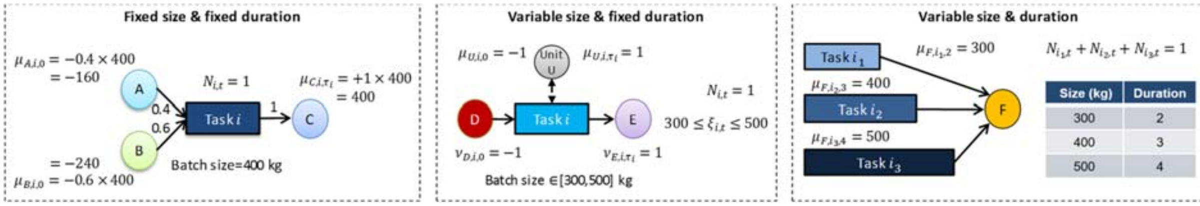


Fig. 6. Values of structural parameters for modelling tasks with fixed/variable sizes and durations with a RTN discrete-time formulation.

Eq. (14) is known to be very tight, and together with the objective function, may be the only constraint needed to tackle a scheduling problem with a discrete-time formulation. Despite its simplicity, the modeler still needs to find the proper set of tasks and resources, as well as the values of the structural parameters, which may be non-trivial for non-conventional problems.

Variables $N_{i,t}$ can be defined as integer instead of binary, to reduce the problem size when dealing with operations that can be executed in one of multiple equivalent equipment units (by equivalent, we mean the exactly same resource requirements and duration, avoiding the need for disaggregating operations into tasks, recall assumption in section 2.3).

2.5.1. Single grid continuous-time formulation

While it is possible to handle variable processing times in a continuous manner with a discrete-time formulation without introducing additional binary variables (Maravelias, 2005), the equipment will remain idle until the next time point in the grid. As a consequence, suboptimal solutions may result, particularly when minimizing makespan. This is no longer the case for continuous-time formulations. We now briefly discuss important aspects of the single grid formulation of Castro et al. (2004) that can handle batch and continuous tasks and be used for multistage adjustable robust optimization (Lappas and Gounaris, 2016).

Since the location of time points in the grid is unknown a priori, extent variables gain a time index to also identify the ending point $t' > t$: $N_{i,t,t'}$ and $\xi_{i,t,t'}$. Then we need to ensure that the time difference ($L_{t,t'}$) between the starting and ending points is greater than the duration of the task (one could also use an equality but that could increase the number of event points required to find the global optimal solutions, a parameter strongly affecting computational performance). For a batch task $i \in I^b$, it will be computed as the sum of a constant α_i plus a size-dependent term $\beta_i \xi_{i,t,t'}$. The disjunction in Eq. (15) enforces the appropriate constraints if the task is active and relaxes them if it is not. Its convex-hull reformulation generates Eqs. (16) and (17).

$$\begin{bmatrix} N_{i,t,t'} \\ v_i^L \le \xi_{i,t,t'} \le v_i^U \\ L_{t,t'} \ge \alpha_i + \beta_i \xi_{i,t,t'} \end{bmatrix} \veebar \begin{bmatrix} \neg N_{i,t,t'} \\ \xi_{i,t,t'} = 0 \\ L_{t,t'} \ge 0 \end{bmatrix} \forall i \in I^b, t, t' > t \tag{15}$$

$$v_i^L N_{i,t,t'} \le \xi_{i,t,t'} \le v_i^U N_{i,t,t'} \; \forall i, t, t' > t \tag{16}$$

$$L_{t,t'} \ge \alpha_i N_{i,t} + \beta_i \xi_{i,t,t'} \; \forall i, t, t' > t \tag{17}$$

When dealing with continuous tasks $i \in I^c$, the focus is on ensuring that the processing rate is within the minimum ρ_i^L and maximum values ρ_i^U and it can be assumed without loss of generality that tasks last a single slot (the optimization can decide to split a continuous operation into as many tasks as needed). Hence, we return to single time indexed variables. One can consider tasks individually, similarly to Eq. (15), but a better approach (stronger linear relaxation and orders of magnitude speed ups) is to combine all tasks $i \in I_j$ that are executed on equipment unit j. It translates into the disjunction in Eq. (18), where h is a parameter representing the time horizon. As shown in Mostafaei and Castro (2017) for pipeline scheduling (see also second paragraph in section 5 of Castro and Mostafaei (2017)), the convex hull reformulation of Eq. (18) leads to Eqs. (7) and (19), which are equivalent to the ones proposed by Castro et al. (2004).

$$\bigvee_{i \in I_j} \begin{bmatrix} N_{i,t} \\ v_i^L \leq \xi_{i,t} \leq v_i^U \\ \frac{\xi_{i,t}}{\rho_i^U} \leq L_t \leq \frac{\xi_{i,t}}{\rho_i^L} \end{bmatrix} \underline{\bigvee} \begin{bmatrix} N_t^{no\ i} \\ \xi_{i,t} = 0\ \forall i \in I_j \\ 0 \leq L_t \leq h \end{bmatrix} \quad \forall j, t \tag{18}$$

$$\sum_{i \in I_j} \frac{\xi_{i,t}}{\rho_i^U} \leq L_t \leq \sum_{i \in I_j} \frac{\xi_{i,t}}{\rho_i^L} + h \cdot (1 - \sum_{i \in I_j} N_{i,t})\ \forall j, t \tag{19}$$

The structural parameters for the continuous-time formulation account for discrete interactions that occur either at the beginning ($\mu_{r,i}$, $\nu_{r,i}$) or end of the task ($\bar{\mu}_{r,i}$, $\bar{\nu}_{r,i}$), and continuous interactions throughout its execution ($\lambda_{r,i}$). They appear in the excess resource balances in Eqs. (20) and (21), where $R_{r,t}^{end}$ represents the amount of resource r immediately before the end of slot t ($R_{r,0}^{end}$ gives the initial resource availability) and $\xi_{i,t,t+1} \equiv \xi_{i,t}$ for continuous tasks. Notice that discrete interactions $\pi_{r,d}$ at known discrete points d need now to be linked to binary variables $Y_{t,d}$ that link d to time point t.

$$R_{r,t} = R_{r,t-1}^{end} + \sum_i [\sum_{t'>t} (\mu_{r,i} N_{i,t,t'} + \nu_{r,i} \xi_{i,t,t'}) + \sum_{t'<t} (\bar{\mu}_{r,i} N_{i,t',t} + \bar{\nu}_{r,i} \xi_{i,t',t})] + \sum_d \pi_{r,d} Y_{t,d}\ \forall r, t \tag{20}$$

$$R_{r,t}^{end} = R_{r,t} + \sum_{i \in I^c} \lambda_{r,i} \xi_{i,t}\ \forall r, t \tag{21}$$

2.6. STN and RTN similarity to UOPSS

A similar paradigm for the representation of advanced planning and scheduling systems is the unit-operation-port-state superstructure (UOPSS) (Zyngier and Kelly, 2012). It is part of Honeywell's Production Scheduler commercial product since 2002 (Kelly, 2003) and has over 45 industrial clients spanning several industrial sectors. UOPSS is similar to a process flowsheet but with information on operating modes.

As an example, consider the continuous multiproduct fruit-juice plant in Fig. 7. The raw-materials for the three types of juices are spring water, apple, pear and grape concentrates, and either carton or bottle packages can be used. In this case, the triangles correspond to the STN material states, while the blender and packaging lines are the equipment resources of the RTN. Above, there is no information about the processing tasks needed to make the different juices. The alternative modes of operation can be seen below, with blender mode GPA consuming all raw-materials to produce the grape blend, mode GP consuming all but the apple concentrate to make the grape juice blend, and mode G producing pure grape juice. Besides the connectivity, the production recipe specifies how much of each component is needed, leading to the values of the structural parameters. The three alternative blending modes on the vertical correspond to the STN/RTN tasks that compete for the blender equipment. Analogously, there are three operating modes for the carton packaging line and three for the bottle line, one for each type of juice.

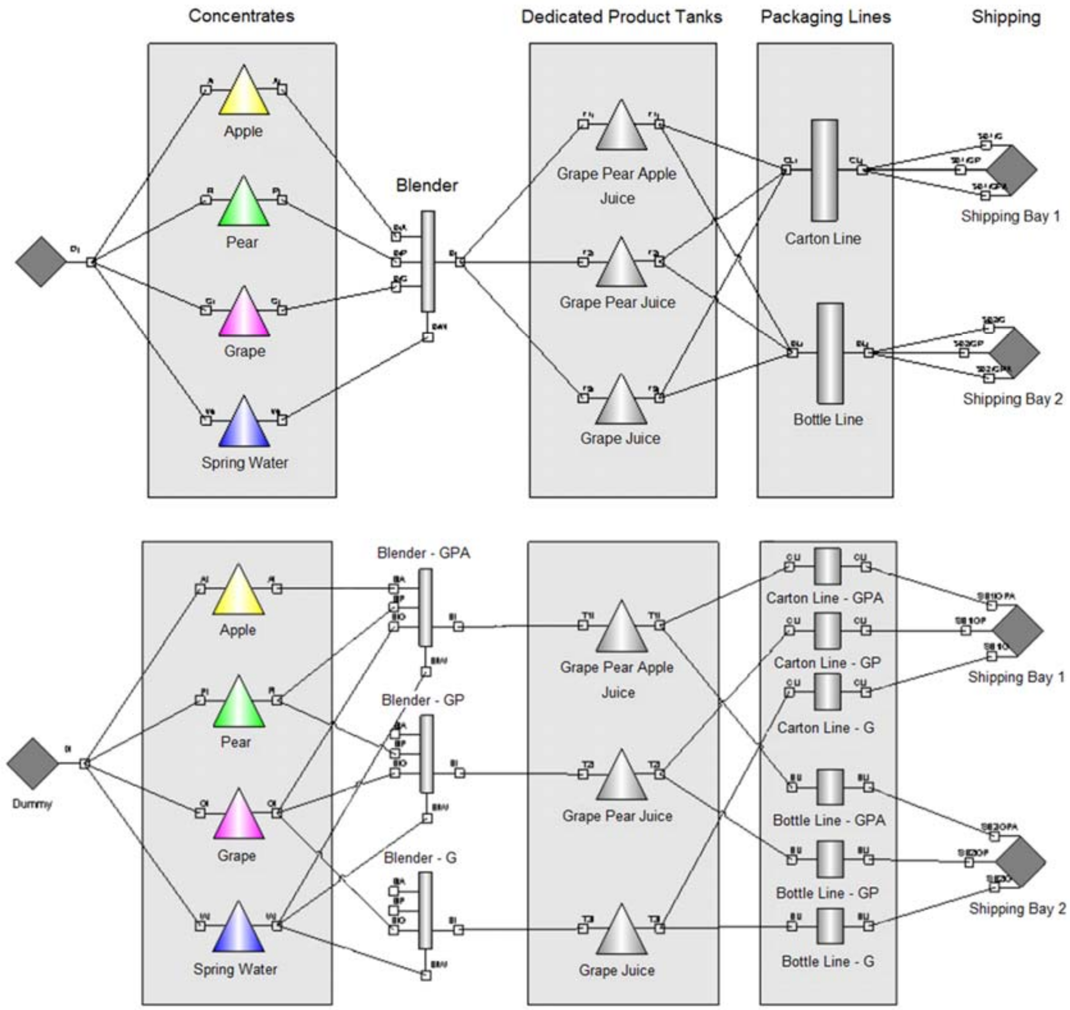


Fig. 7. Unit-operation-port-state superstructure (UOPSS) for a fruit juice plant. The representation of the process flow diagram is above. Below, we can see the alternative operating modes for the blender and packaging lines.

The drawback of representing all the operating modes in UOPSS is that complex production recipes may be difficult to visualize. More so for the RTN, which also shows the equipment units on the diagram and their connectivity to tasks. The STN is the simplest to visualize.

3. Beyond conventional scheduling

In many scheduling problems reported in the literature, interactions between tasks are rather limited. We can find tasks competing for key resources (e.g. equipment), and in the case of multistage production recipes, tasks executed sequentially. In the three industrial case studies discussed in this section, task interaction is more extensive, making it much harder to develop an effective model for decision making. While the focus is on showing how Generalized Disjunctive Programming and the Resource-Task Network can help us cope with the modelling challenge, other important aspects are highlighted.

In the problem in section 3.1, some of the tasks require heating or cooling. The simplest option is to rely on external utilities but it is more economical and environmentally friendly to carry out heat integration. The latter requires tasks to be synchronized, with the model allowing a task' heating/cooling needs to be divided over multiple exchangers to maximize energy efficiency, while accounting for energy balances and temperature driving forces. This is a bi-objective problem with a clear tradeoff between makespan and utility consumption.

Section 3.2 deals with the transportation of liquid products by pipeline. Multiple products may exist in a segment at a certain time, being critical to know their precise location to trigger entering and exit events. An entering product interacts with all products inside by forcing them to move (plug flow). The modeling challenge is that these are not known a priori since volumes and sequence of injections are to be determined by the optimization.

Conventional scheduling problems have fixed recipes but sometimes the value is in the recipe optimization. It is the case of crude oil and refined products blending in section 3.3, which also explicitly deal with transportation, in this case, between dedicated and blending tanks. The latter introduces bilinear terms, transforming the model from an MILP to a non-convex MINLP that can be rigorously solved to global optimality using an MILP-NLP decomposition strategy in which the MILP is generated from piecewise relaxation techniques. We discuss two alternative MINLP formulations, showing that a larger size with more bilinear terms may lead to a better computational performance.

3.1. Simultaneous scheduling and heat integration

Increasing energy efficiency in industry is one of the objectives of the 2014-2018 strategic plan of the United States Department of Energy and of the European Union framework programme for research and innovation (Horizon 2020). It can be achieved by better energy management schemes, derived from system optimization, of which heat integration is a well-known example (Linnhoff et al., 1982; Biegler et al., 1997). The basic idea is to reduce the consumption of external utilities (e.g. steam, cooling water) by properly exchanging heat between hot and cold process streams. The focus has been primarily on energy intensive continuous plants, with more recent efforts increasingly dealing with batch plants (Fernández et al., 2012). For effective heat integration in batch plants, both the temperature levels of streams and their timing of occurrence need to be right (Papageorgiou et al., 1994; Adonyi et al., 2003; Halim and Srinivasan, 2009; Chen and Chang, 2009; Seid and Majozi, 2014). We now focus on direct heat integration, which does not consider energy storage, briefly describing how to tackle the simultaneous scheduling and heat integrated problem (Castro et al. 2015; Castro 2016b).

In a vegetable oil refinery, the production recipe of product p is divided into subtasks i of known duration $d_{p,i}$, see Fig. 8. Some tasks require heating (cold streams, in blue) or cooling (hot streams, in red), from an initial t_{ps}^{in} to a target temperature t_{ps}^{out}. Hot and cold process streams ps have heat capacity cp_{ps} and can only exchange energy if originating from products not executed on the same unit. Note that product-unit assignments and batch sizes are fixed and a single batch is required per product.

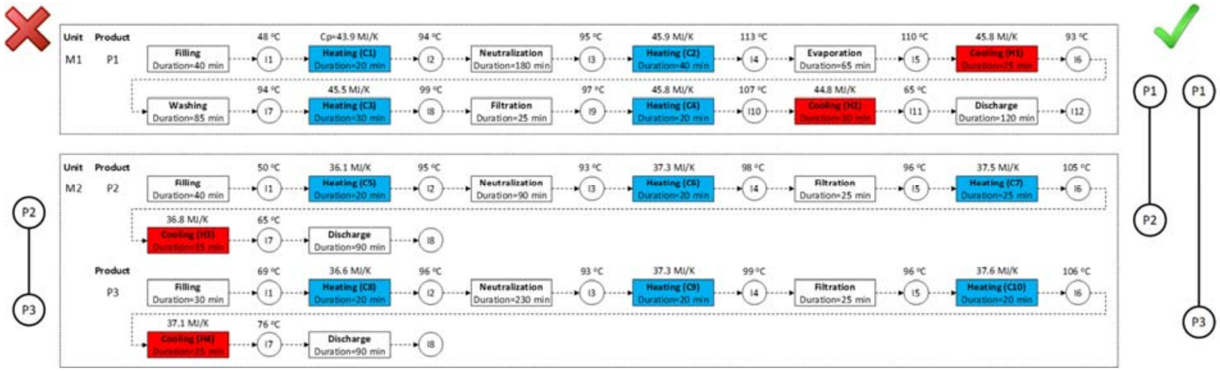


Fig. 8. Production recipe of three products (P1-P3) with fixed unit assignments (P1→M1, P2→M2, P3→M2). Heat integration only allowed between process streams linked to subtasks executed in different units.

One can consider two separate blocks for generating the model for simultaneous scheduling and heat integration. On the top left of Fig. 9, we consider the general precedence model of Méndez et al. (2001) for determining the sequencing of all pairs of products p and p' executed in unit m, and the timing of all tasks. Notice that tasks are considered in an aggregate manner, with the left term in the disjunction, active whenever p precedes p' ($X_{p,p'} = 1$), relating the ending time of the last task of product p ($E_{p,|I_p|}$) with the starting time of the first task of product p' ($S_{p',1}$). Then, the ending time of a subtask is equal to the starting time plus the duration, and the starting time of task $i + 1$ is equal to the ending time of i (no idle times allowed between subtasks, which would change the temperature values). The makespan variable MK is the ending time of the final task of the last product in the schedule.

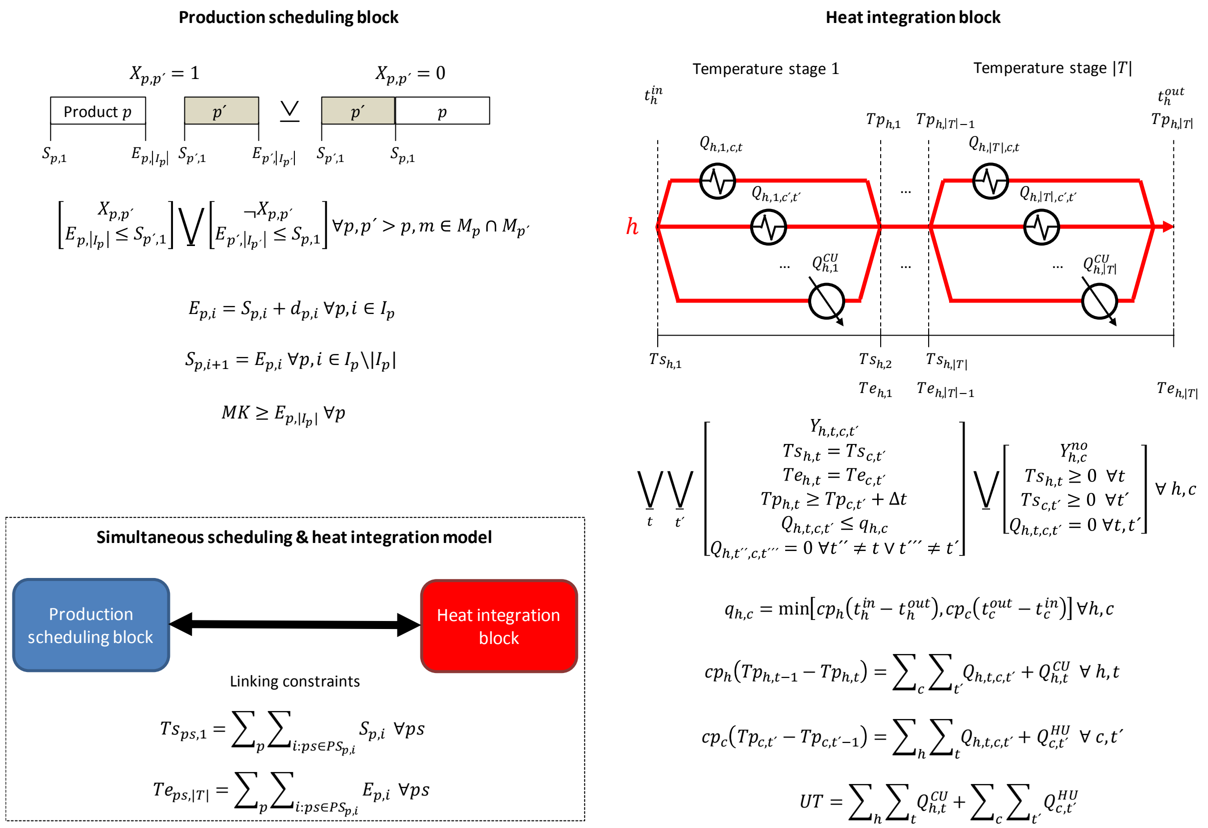


Fig. 9. Scheduling model for heat integrated batch plants is built from two separate blocks of constraints. Heat integration block allows for multiple ($|\boldsymbol{T}|$) temperature-changing stages.

On the right of Fig. 9, we show the heat integration block. The superstructure of the cooling possibilities for hot stream h is on the top right. It features $|T|$ temperature changing stages with matches in parallel and isothermal mixing (to avoid bilinear terms in the energy balances), following the work of Yee et al. (1990). On stage t, the stream can exchange energy with all process cooling streams and the external cold utility ($Q_{h,t}^{CU}$). The temperature at the end of the stage is given by variable $Tp_{h,t}$ and we are assuming that $Tp_{h,0} = t_h^{in}$ and $Tp_{h,|T|} = t_h^{out}$. Starting $Ts_{h,t}$ and ending time $Te_{h,t}$ variables are also needed, with $Ts_{h,t+1} = Te_{h,t}$.

The energy exchange between hot stream h in stage t and cold stream c in stage t', is identified through binary variable $Y_{h,t,c,t'} = 1$. On the disjunction on the middle-right of Fig. 9, there are five constraints to enforce if the match is selected. The first two equal the starting and ending times, while the third ensures that the temperature difference between the two streams is at least the minimum approach temperature Δt (note that the hot/cold stream is at its lowest/highest temperature at the end of the stage). The energy exchange $Q_{h,t,c,t'}$ is bounded by upper bound $q_{h,c}$ and the fifth constraint prevents the two streams from interacting with each other in other stages.

On the bottom-right of Fig. 9, below the disjunction, we show the formula to compute parameter $q_{h,c}$ and the energy balances. The first two balances state that the change in energy for a hot/cold stream in a temperature-changing stage, computed by multiplying the heat capacity by the temperature difference, is equal to the sum of all energy exchange variables with cold/hot process streams and utilities. The third balance gives the total utility consumption variable UT.

The models considered in the scheduling and heat integration blocks use different sets of timing variables, either task of stream related. Since the correspondence between the two is known from the production recipe ($PS_{p,i}$ holds the process stream corresponding to recipe-stage i of product p), we just need to add the linking constraints on the bottom-left of Fig. 9, to generate the integrated model.

One interesting aspect of the simultaneous scheduling and heat integration problem is that it is bi-objective, since the more the time given for a task to wait for its perfect match (longer makespan), the lowest the total utility consumption compared to the case of no integration (100%). This is apparent from the Pareto frontiers in Fig. 10 that also show higher savings when postulating $|T|$=3 temperature-changing stages. The minimum makespan is 805 min, which corresponds to 81.8% of utility consumption. To achieve the minimum consumption of 60.8%, major delays in production time are needed (1365 min). A good compromise is to set the production time to 890 min, for which the optimal energy consumption is 73.8%, see optimal schedule and heat-exchange matches in Fig. 11.

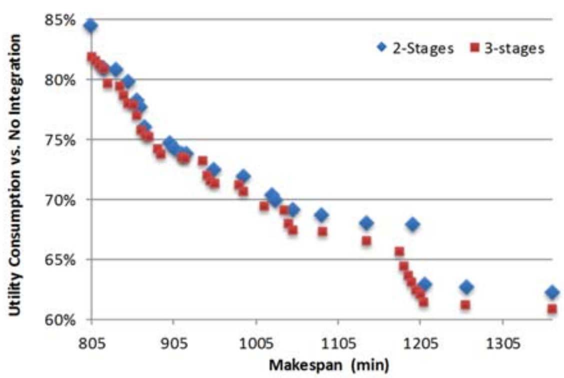


Fig. 10. Tradeoff between utility consumption and makespan as a function of number of temperature-changing stages for a 26 streams example.

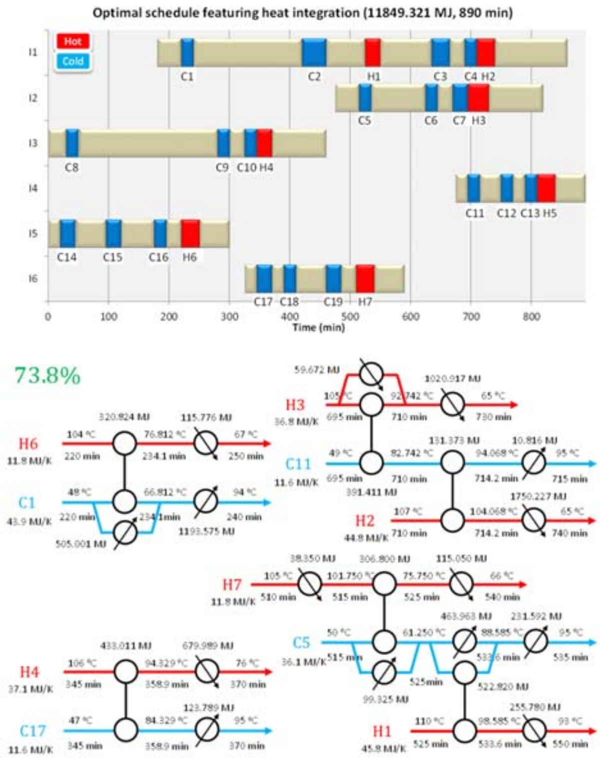


Fig. 11. A practical Pareto optimal solution for a 26 streams example.

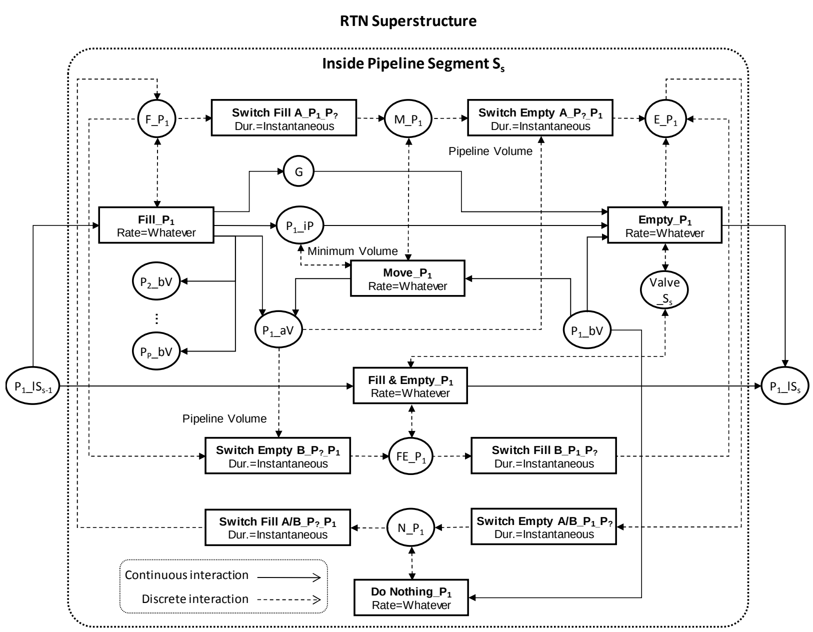
3.2. Pipeline scheduling

In the petroleum supply chain, pipelines are used for carrying crude oil from wells to refineries and for sending refined products (e.g. gasoline, diesel, liquified petroleum gases) from refineries to harbors and distribution centers serving local markets. The main goal of multiproduct liquid pipeline operators is to keep tank levels at regional farms within acceptable ranges to be able to meet demand on time. Pipeline scheduling involves deciding the amount of each product going from an input to an output node, as well as the sequence and timing of injections and deliveries. System configuration ranges from straight pipelines with a single input and output node, to multilevel treelike and mesh structures with multiple dual-purpose nodes, sometimes allowing for reversible flow.

Mathematical formulations for pipeline scheduling have received a great deal of attention over the last 15 years and have used discrete and continuous representations of time and volume. Discrete representations in both time and volume work with a fixed flowrate and greatly facilitate the modeling of the presence of different batches of the same product inside a segment (Hane and Ratliff, 1995; Rejowski and Pinto, 2003; Magatão et al., 2004; Herran et al., 2010). Yet, most recent work has focused on continuous representations to allow for variable flowrates and improve computational performance (Cafaro and Cerdá, 2004; 2011; 2012; 2014; Relvas et al. 2006; Castro, 2010; MirHassani and Jahromi, 2011; Mostafei et al. 2015; Mostafaei and Castro 2017; Castro and Mostafaei, 2017). These can be divided into batch- and product-centric approaches. The former requires the user to characterize the initial pipeline contents in terms of batches and to postulate their total number, decisions that may compromise solution quality and feasibility. The latter does not allow multiple batches of a product inside a segment (assumption implicit in the following), which may be required to ensure feasible schedules when dealing with segments covering long distances.

The key for a general formulation is to master the modeling of what goes on inside a segment, the part of the pipeline connecting consecutive nodes. If we assume that the segment is active, a product can be affected in five different ways: (i) when the product is not inside, neither the volume nor location change; (ii) when the product is entering the segment while another product is leaving on the other side, its volume increases; (iii) the volume remains constant when the product is entering on one side and leaving on the other side; (iv) the volume remains constant but the location changes when the product is already inside but is neither leaving nor entering the segment; (v) the volume decreases when the product is leaving the segment due to the entering of another product.

Castro (2010) proposed a rather complex RTN superstructure to model a pipeline segment that is reproduced on the left of Fig. 12. It took one month to develop and features: continuous fill, move and empty tasks of the continuous type; batch tasks to switch the product being filled/emptied and prevent the appearance of product sequences that are forbidden; three different types of material resources per product, one holding the volume inside and two related to its location; five different types of equipment resources per product, one for each of the five modes of operation just described.



GDP Formulation

$$\veebar_{p}\begin{bmatrix} Y_{s,p,t}^{S,in} \\ LC_{s,p,t}=0 \\ f_s^{S,L} \le F_{s,p,t}^{S,in} \le f_s^{S,U} \\ \dfrac{F_{s,p,t}^{S,in}}{\rho_s^{S,U}} \le L_t \le \dfrac{F_{s,p,t}^{S,in}}{\rho_s^{S,L}} \end{bmatrix} \veebar \begin{bmatrix} Y_{s,t}^{S,no\ i} \\ LC_{s,p,t} \le v_s^S \ \forall p \\ F_{s,p,t}^{S,in}=0 \ \forall p \\ L_t \le h \end{bmatrix} \ \forall s,t$$

$$\veebar_{p}\begin{bmatrix} Y_{s,p,t}^{S,out} \\ RC_{s,p,t}=v_s^S \\ f_s^{S,L} \le F_{s,p,t}^{S,out} \le f_s^{S,U} \end{bmatrix} \veebar \begin{bmatrix} Y_{s,t}^{S,no\ o} \\ RC_{s,p,t} \ge 0 \ \forall p \\ F_{s,p,t}^{S,out}=0 \ \forall p \end{bmatrix} \ \forall s,t$$

$$LC_{s,p,t}^{end} = LC_{s,p,t} + \sum_{p' \ne p} F_{s,p',t}^{S,in} \ \forall s,p,t$$

$$RC_{s,p,t}^{end} = RC_{s,p,t} + \sum_{p' \ne p} F_{s,p',t}^{S,out} \ \forall s,p,t$$

$$LC_{s,p,t} = LC_{s,p,t-1}^{end} - ZC_{s,p,t} \ \forall s,p,t$$

$$RC_{s,p,t} = RC_{s,p,t-1}^{end} - ZC_{s,p,t} \ \forall s,p,t$$

$$\begin{bmatrix} Y_{s,p,t}^{S,is} \\ V_{s,p,t}^S \le v_s^S \\ ZC_{s,p,t}=0 \end{bmatrix} \veebar \begin{bmatrix} \neg Y_{s,p,t}^{S,is} \\ V_{s,p,t}^S = 0 \\ ZC_{s,p,t} \le f_s^{S,U} \end{bmatrix} \ \forall s,p,t$$

Fig. 12. Alternative ways of modeling a pipeline segment handling multiple incompressible products in systems with unidirectional flow.

More recently, Castro and Mostafaei (2017) used GDP to develop an equivalent continuous-time formulation that reduced problem size by a factor of 3-5 and improved computational performance by at least one order of magnitude, see Table 3. On the right of Fig. 12, we show the most important constraints. The top two disjunctions ensure that, if segment s is active, a single product p enters ($Y_{s,p,t}^{S,in} = 1$) and leaves ($Y_{s,p,t}^{S,out} =$

1) during slot t. Notice that the first set of constraints inside the disjunctive terms feature left ($LC_{s,p,t}$) and right ($RC_{s,p,t}$) coordinate variables, which locate the product inside the segment at the start of slot t. More specifically, and assuming unidirectional flow from left to right, a product can only enter if its left coordinate is equal to zero, and can only leave if its right coordinate is equal to the segment volume v_s^S. The two coordinate variables, together with the product volume inside ($V_{s,p,t}^S$), trigger events in a similar manner to the three material resources of the RTN superstructure. The remaining challenge is to ensure that coordinates only increase for the products that are inside the segment.

Table 3. Computational performance of product-centric formulations in problems from straight pipeline systems (best performer in bold, maximum computational time in italic).

Model	RTN-based					GDP-based				
Problem	Binary Variables	Total Variables	Equations	Obj (h)	CPUs	CPUs	Obj (h)	Binary Variables	Total Variables	Equations
Ex1	4763	9412	6155	No solution	*18000*	**7884**	213.94	869	3012	4604
Ex2	4488	8945	6032	126.19	16320	**111**	126.19	880	2959	4560
Ex4	2814	6028	4158	145.00	110	**6.22**	145.00	546	1998	3091
Ex5	1358	3357	2712	158.33	16.8	**1.85**	158.33	385	1309	2085
Ex6	1182	2958	2405	114.29	2.66	**0.51**	114.29	336	1148	1815
Ex7	1960	5324	4371	46.00	137	**4.14**	46.00	560	2056	3648

Four sets of volume balances are used for this purpose, see middle right of Fig. 12. Despite featuring only continuous variables, they are somewhat like the excess resource balances in Eqs. 20-21. Variables $LC_{s,p,t}^{end}$ and $RC_{s,p,t}^{end}$ give the coordinate values at the end of slot t and increase whenever a different product (p') is entering/leaving the segment during t, regardless of p being inside or not. We then allow the products not inside the segment ($Y_{s,p,t}^{S,is} = 0$) to reset their coordinates by allowing variables $ZC_{s,p,t}$ to be greater than zero. The outcome are the coordinates at the start of the next slot.

In Fig. 13, we show the schedule that minimizes makespan for a specific straight pipeline system (Castro and Mostafaei, 2017). It was obtained by the GDP-based formulation. Notice that segments are never idle and that the flowrate changes between segments and with time.

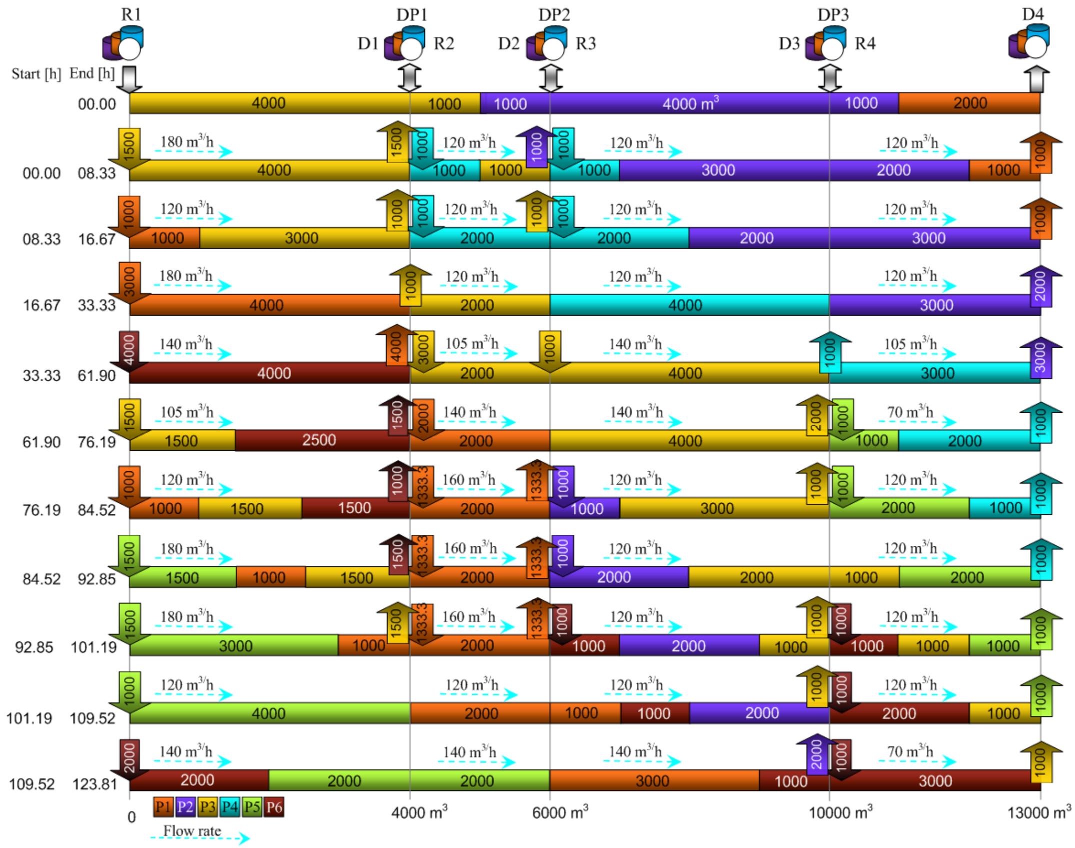


Fig. 13. Optimal schedule for a straight pipeline system with two terminal nodes (input node R1 and output node D4) and three intermediate multipurpose nodes (DP1-DP3). Arrows show injected/withdrawn volumes (m^3) during a time slot, while the volumes inside the pipeline segments are the product volumes at the end of the slot.

Overall, the main advantage of the GDP-based approach is its simplicity, allowing it to be easily extendable to systems with reversible flow (Castro, 2017b).

3.3. Crude oil and refined products blending

The scheduling of blending operations is a very important problem in petroleum refineries. In crude oil blending, different crudes are mixed in a way that minimizes cost while meeting quality constraints in the charging tanks that feed the crude oil distillation unit(s). In the system by Lee et al. (1996), incoming crude marine vessels have estimated times of arrival, feeding dedicated (see Fig. 14) or blending storage tanks, one at a time. In the system of Li et al. (2007), a vessel can carry multiple crudes that are sent directly to charging tanks.

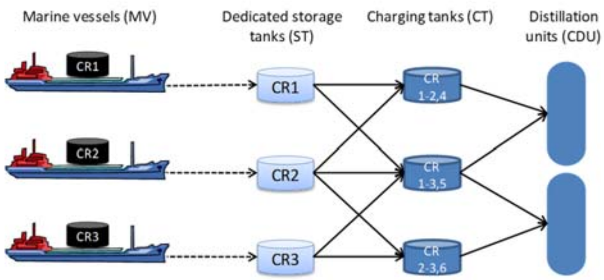


Fig. 14. Crude oil blending in petroleum refineries.

3.3.1. Modeling logistic constraints

Both systems have important logistic constraints that restrict certain combinations of unit connections. These involve binary variables and thus can be formulated using logic propositions (Castro, 2016). An example is the restriction of no simultaneous inlet and outlet flow to a charging tank, see Fig. 15 on the left, where binary variables $Y_{u,u',t} = 1$ indicate if the connection between units u and u' is active during slot t, and $Y_{u,t}^{no} = 1$ identify idle tanks. The RTN way of modeling can be seen on the right of Fig. 15, and requires the definition of auxiliary transfer tasks (two) and resources, four material and two equipment resources (Castro and Grossmann, 2014). Notice that: (i) unitary equipment resource CT1 prevents the execution of “Transfer in CT1” and “Transfer CT1,CDU” at the same time; (ii) material resources of type “1_CT1” represent the material inside the tank, needed to model inventory constraints, while “1_CT$^{i}_{1}$” represents crude CR1 at the entrance of CT1, a location resource r that is immediately consumed when produced by making $R_{r,t} = 0\ \forall t$, (location resources appear for the first time in Barbosa-Póvoa and Macchietto, 1994); (iii) variable recipe tasks are needed to allow one or two materials to be transferred simultaneously, with the relative proportion to be determined by the optimization; (iv) variable recipe blending tasks consume materials in the exact same proportion as in their origin tank. Overall, the GDP-based approach is simpler and leads to a smaller mathematical problem.

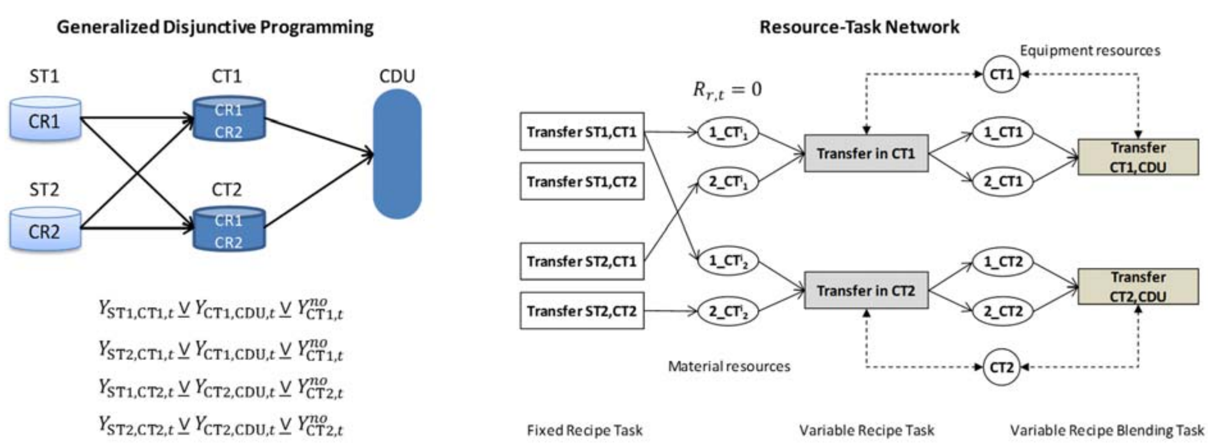


Fig. 15. Alternative ways of modeling logistic constraints.

3.3.2. Batch vs. continuous blending

In a gasoline blending system, components from dedicated supply tanks are mixed in in-line blenders or blending tanks. In-line blenders operate in a continuous manner, switching, from time to time, from one grade of gasoline (e.g. regular, medium, premium) to another (Glismann and Gruhn 2001; Li and Karimi, 2011; Li et al. 2016; Castillo-Castillo and Mahalec, 2016). The blended material goes either to a product tank (see Fig. 16 on the left) or directly to the pipeline. Blending in tanks resembles batch operation since tanks, as previously described, can either have flow in or out (see Fig. 16 on the right). Examples can be found in Kolodziej et al. (2013), Castro (2015). and Lotero et al. (2016), where the multiperiod operation is an extension of the well-known pooling problem (Haverly,1978; Ben-Tal et al. 1994; Tawarmalani and Sahinidis, 2002; Meyer and Floudas, 2006; Alfaki and Haugland, 2013; Boland et al. 2016).

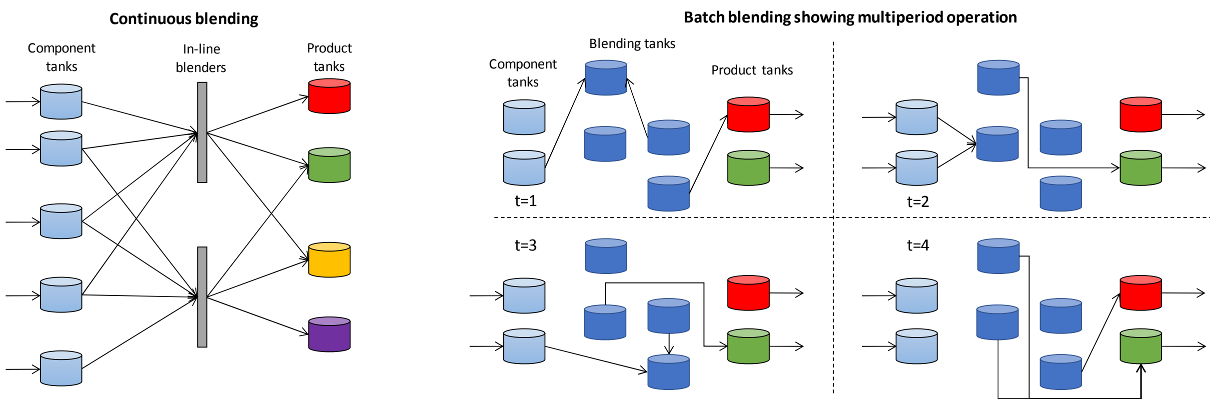


Fig. 16. Continuous vs. batch gasoline blending.

A critical aspect of scheduling blending operations is the determination of blending recipes, i.e. the amounts of components to mix such that the products quality properties respect given lower and upper bounds. It is often assumed that components blend linearly, either on a volumetric or weight basis, despite not being the case for research octane number, motor octane number and Reid vapor pressure (Singh et al., 2000). Considering nonlinear blending rules for such properties can increase the accuracy of the solution and reduce quality giveaways.

Under the assumption of linear blending rules, the scheduling of continuous blending operations is, in essence, an MILP, since the properties of the component tanks are known data. Nevertheless, it is formulated as an MINLP due to the necessity of enforcing constant processing rates when producing the same gasoline grade over consecutive time slots. It forces the definition of in-line blenders processing rates as model variables rather than just ensuring that they are within minimum and maximum values (recall Eq. 19), the standard for continuous plant scheduling. In contrast, the properties inside blending tanks will be determined by the optimization and so the scheduling of batch blending operations needs to be formulated as an MINLP.

3.3.3. Alternative formulations

Work for the single period pooling problem has brought us alternative NLP formulations that vary in the type of variables being used. Some lead to a stronger linear relaxation that, together with a smaller problem size, increase the likelihood of proving global optimality. Multiperiod operation turns the problem into an MINLP and two alternative formulations have been used.

On the left of Fig. 17, we show the approach used by most models for process networks (Kolodziej et al., 2013). Variables are related to the flows between tanks ($F_{u,u',t}$), compositions ($C_{q,u,t}$) and tank volumes ($V_{u,t}$). These are shared with the p-formulation by Haverly (1978). The circles indicate incoming crudes (1-2) and crudes initially in the system (3-6), with known compositions $c_{q,u}$, and the final products (A, B). Below the network, we show the volume balances for every quality q, tank u and slot t. Notice that all terms involve the product of two continuous variables, bilinear terms that are non-convex.

On the right of Fig. 17, we have a source-based approach that deals only with liquid fuels l of known compositions $c_{q,l}$. In every tank, there are volume variables $V_{l,u,t}$ for every liquid fuel, which can be viewed as material location resources of the RTN (identity kept throughout the system, only the location changes). The volume balances are written for every liquid l instead of quality q and have the advantage of being linear. We do need to enforce that the composition in all outlet streams from a tank matches the composition inside for all qualities q. This is equivalent to saying that the volume of liquid l leaving tank u to tank u' ($F_{l,u,u',t}$) is a fraction ($X_{u,u',t} \in [0,1]$) of the volume inside, which leads to a bilinear term. Note that to know the total volume inside a tank and the total flow between tanks, one needs simply to sum the individual variables.

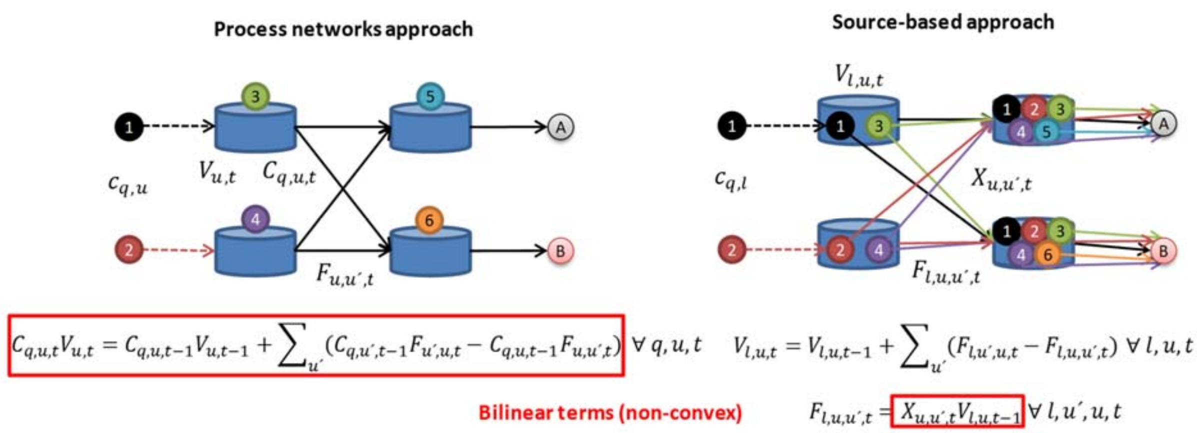


Fig. 17. Alternative formulations for multiperiod batch blending problems

These two alternative formulations are equivalent in the sense that there exists a one-to-one correspondence between the feasible sets that preserves the objective function, but differ in the quality of the relaxation. Lotero et al. (2016) have shown that the source-based formulation is tighter, often leading to a better computational performance when solving the MINLPs to global optimality with BARON (see results in Table 4, from Castro, 2015). The higher-quality relaxation also explains why the DICOPT solver, which works under the assumption that the MINLP is convex (which is not the case), finds feasible solutions in four problems instead of one.

Notice that the formulations share the number of binary variables, with the source-based model featuring a larger number of bilinear terms and significantly more variables (disaggregated volumes) and constraints.

Table 4. Comparison between a process networks and a source-based approach for multiperiod batch blending problems.

Model	Process networks-based					Source-based				
Problem	Variables	Equations	Bilinear Terms	DICOPT feasible?	BARON CPUs	BARON CPUs	Variables	Equations	Bilinear Terms	DICOPT feasible?
029	103	202	64	No	3.13	**0.33**	219	294	64	No
146	223	617	256	No	1227	**91.3**	689	965	480	Yes
480	313	879	376	No	**302**	453	941	1383	720	No
531	273	732	358	No	97.4	**43.3**	878	1218	684	No
718	223	603	244	Yes	**3.56**	3.97	672	939	456	Yes
721	223	623	256	No	265	**21.4**	689	971	480	Yes
852	305	859	376	No	231	**134**	933	1363	720	Yes

3.3.4. MILP-NLP solution strategy for mixed-integer bilinear problems

We have just seen that multiperiod blending problems feature binary variables and non-convex bilinear terms of type $x_i x_j$. Non-convex MINLPs can be solved to global optimality using commercial solvers like BARON (Tawarmalani and Sahinidis, 2005) and ANTIGONE (Misener and Floudas, 2013) that rely heavily on spatial branch-and-bound. The alternative for the special class of mixed-integer bilinear programs is to rely on an MILP-NLP decomposition strategy (Jia et al., 2003).

The MILP relaxation can be derived on a variety of ways. The simplest, is to replace the bilinear terms with a new set of continuous variables $z_{i,j} = x_i x_j$. The most common, is to add the McCormick envelopes of the bilinear function (McCormick, 1976), which consist of four sets of inequality constraints relating new variables $z_{i,j}$ to original variables x_i and x_j, and their lower and upper bounds. The relaxation can be made tighter by using partition-dependent bounds (Bergamini et al., 2005; Karuppiah and Grossmann, 2006; Meyer and Floudas, 2006; Wicaksono and Karimi, 2008; Castro, 2015b). The so-called piecewise McCormick envelopes typically rely on uniform and univariate (x_j) partitioning, see illustration in Fig. 18, where the top four constraints inside the disjunction are the McCormick envelopes. Notice that binary variables $Y_{j,n}$ are added to the problem to identify the optimal partitions.

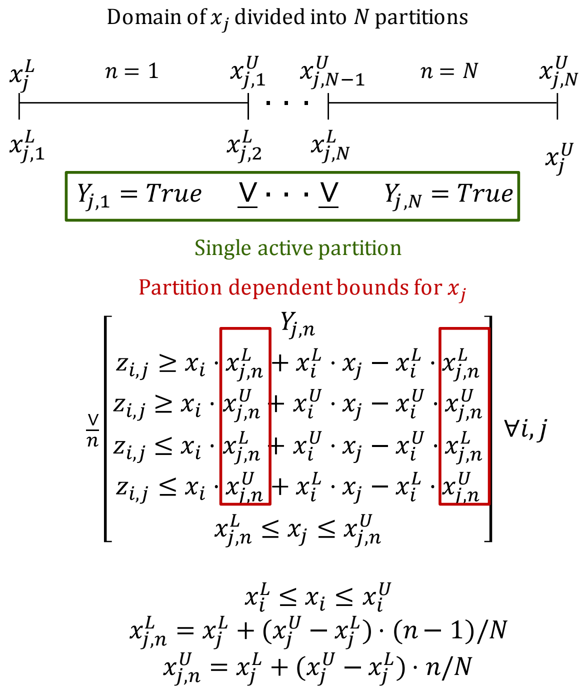


Fig. 18. Piecewise McCormick envelopes

Piecewise McCormick reduces to the bilinear envelopes for $N = 1$ and it is recommended for $N = \{2, \ldots, 9\}$. Beyond that, we recommended using the equivalent relaxation from multiparametric disaggregation (Teles et al., 2013; Kolodziej et al., 2013b, Castro, 2016b), which has the advantage of the number of added binary variables scaling logarithmically with the number of partitions (vs. linearly). In such case, the number of partitions needs to be a power of ten, i.e. $N = \{10,100,1000, \ldots\}$.

Piecewise relaxation approaches do not need to be integrated with spatial branch-and-bound procedures, since global optimality is guaranteed as $N \rightarrow \infty$. This is because the tighter the relaxation, the closer the solution from the MILP is to the feasible region of the original MINLP. The values of the model variables can thus be used as initialization points. More specifically, the binary variables of the original MINLP can be fixed, reducing it to a NLP, which can be quickly solved by a local solver. Assuming a feasible MINLP problem with an objective function being minimized, the value of the objective function from the MILP relaxation provides the lower bound, while the objective from the NLP (if fixing the binary variables did not compromise feasibility) provides the upper bound. An optimality gap can thus be computed.

3.3.5. Other sources of bilinear terms

Another source of bilinear terms in scheduling formulations are time-dependent inventory costs. Let variable $\bar{V}_{u,t}$ represent the average inventory in tank u during slot t and let parameter c_u hold the inventory cost ($/bbl/day). If one is using a continuous-time formulation with a single time grid, the total inventory cost is computed as the sum of bilinear terms $c_u \bar{V}_{u,t} L_t$, since the length of the slot (L_t) is going to be determined by the optimization. It is no longer the case for a discrete-time formulation since the duration of all time slots is known a priori (recall section 2.2). The issue is that there are also more slots in a discrete time grid, which is translated into more bilinear terms of the blending type (blending constraints have a time index in their domain, recall Fig. 17).

Results for the crude oil blending problem have shown that the best time representation depends on the objective function, see Table 5 (Castro, 2016). Discrete time is better for cost minimization, despite the number of slots being one order of magnitude larger. Switching to gross margin minimization, a linear objective function, reverses the order. While the differences in solution quality are minor, using fewer slots has the advantage of generating schedules with less frequent changes in operation, which are easier to implement in practice.

Table 5. Best-found solutions when solving crude-oil blending MINLPs resulting from discrete- and continuous-time formulations up to maximum computational time of 3600 CPUs.

Objective	Cost minimization				Gross margin maximization			
Time representation	Discrete		Continuous		Discrete		Continuous	
Problem	Slots	Obj (k$)	Obj (k$)	Slots	Slots	Obj (k$)	Obj (k$)	Slots
P1	97	**209.585**	210.537	8	97	7983	**7985**	4
P2	81	**319.140**	320.496	8	81	10240	**10246**	7
P3	97	**284.781**	287.000	8	49	8542	**8574**	8
P4	121	**319.875**	333.331	7	121	**13258**	**13258**	7

Interestingly, using the two-stage MILP-NLP decomposition procedure with the standard McCormick relaxation for the discrete-time formulation and the objective of cost minimization leads to a zero MINLP-MILP gap. In other words, the solution for the MILP relaxation problem is the global optimal solution of the MINLP. Confirmation is obtained much faster for P1-P3, compared to using BARON and ANTIGONE, see results in Table 6. For P4, both cost and optimality gap are better.

Table 6. Computational performance for MILP-NLP decomposition with McCormick envelopes when solving crude-oil blending problems with discrete-time formulation for the objective of cost minimization.

Problem	Approach	Cost ($)	Gap	CPUs
P1	MILP-NLP	**209585**	0.0000%	72.6
	ANTIGONE	**209585**	0.0001%	1557
	BARON	**209585**	0.0001%	305
P2	MILP-NLP	**319140**	0.0000%	662
	ANTIGONE	319252	10.9%	*3600*
	BARON	**319140**	38.5%	*3600*
P3	MILP-NLP	**284781**	0.0000%	346
	ANTIGONE	**284781**	11.1%	*3600*
	BARON	397208	112%	*3600*
P4	MILP-NLP	**322300**	7.6%	*3600*
	ANTIGONE	No solution	17.6%	*3600*
	BARON	324746	37.9%	*3600*

4. Demand side management

A new area for scheduling that has emerged in recent years is demand side management (DSM), which refers to the active management of electricity demand and has become a major topic in power system operations and economics. Balancing electricity supply and demand is the principal task of any power system. Traditionally, power systems engineers have been mainly concerned with optimizing the supply side for a given demand. However, there is also significant flexibility on the demand side, which can be leveraged to improve grid performance and reliability. This potential in DSM has been recognized in recent years and is now deemed crucial in tackling the challenges of increasing volatility in electricity availability and demand.

For electricity consumers, DSM constitutes the opportunity to benefit from financial incentives by adjusting their electricity consumption. Especially for the chemical industry, which is a major electricity consumer, DSM is becoming increasingly critical for maintaining profitability. Due to the time-sensitive nature of electricity prices and demand response activities, scheduling plays a major role in DSM. In the following, we use the example of operating reserve scheduling to demonstrate how new scheduling and optimization approaches can be used to take advantage of new opportunities arising from DSM. For more details on industrial DSM, we refer to a recent comprehensive review by Zhang and Grossmann (2016).

4.1. Scheduling operating reserve under uncertainty

In the electrical energy market, electricity is traded as a commodity; here, electricity consumers are concerned with the question of when and how much electricity to purchase, which is a difficult problem considering the high fluctuations in electricity price. Electricity can also be traded as a service. In order to balance electricity supply and demand in real time, we rely on backup capacities that can be dispatched within a short amount of time. These backup capacities do not represent actual power generation but rather a guarantee to dispatch when required; hence, they are traded as services in the so-called ancillary services market.

One type of ancillary services is referred to as operating reserve. Operating reserves typically require a response time of 10 to 30 minutes and are needed in relatively large quantities. During a contingency event, the electricity supply in the power system falls below the demand. In this situation, operating reserves can either be dispatched by power generating facilities with fast ramp-up times, which increase their power output to match the demand; or they can be dispatched by electricity consumers that reduce their power consumption, which likewise takes electricity supply and demand back to the same level. The second type of operating reserve is often less expensive and reduces the need for additional power plants. We show examples of both cases in Sections 4.2 and 4.3, respectively.

The major challenge in scheduling operating reserve lies in the uncertainty in the time and amount of dispatch. Operating reserves are needed in times of contingency, which are very difficult to predict (otherwise, there would be no contingency events). This uncertainty can be a huge burden for the operating reserve provider since dispatch of operating reserve can lead to process infeasibilities and shortfalls in other parts of its operations. Nonetheless, dispatch of operating reserve has to be guaranteed when it is requested anytime during the agreed period of time; noncompliance would result in extremely high penalties, or one may not even be allowed to participate in the market. Because of these strict rules, a worst-case approach has to be taken when deciding when and how much operating reserve to provide such that one benefits from the additional revenue, but does not jeopardize vital regular operations.

4.1.1. Budget uncertainty set

A popular worst-case approach is robust optimization (Ben-Tal et al, 2009), which guarantees feasibility for all possible realizations of the uncertainty specified over a given uncertainty set. The uncertainty set has to be chosen carefully as it defines the level of conservatism of the robust optimization problem. In the case of operating reserve, we could apply a simple box uncertainty set defined as follows:

$$U(R) = \{D: 0 \leq D_t \leq R_t \quad \forall\, t \in T\} \tag{22}$$

where U denotes the uncertainty set, R is the amount of operating reserve provided, and D is the actually dispatched operating reserve, which is uncertain. Note that while R and D are vectors, R_t and D_t are the corresponding tth elements, with t being the time index and T denoting the set of time periods.

Obviously, the uncertainty set U is very conservative since D_t could take the value of R_t for all $t \in T$. Instead of U, we propose to apply a budget uncertainty set (Bertsimas and Sim, 2004) that provides the flexibility of adjusting the level of conservatism (e.g. based on historical data). Through a simple transformation, $D_t = R_t\, w_t$, we replace D_t by the new uncertain parameter w_t, which represents the normalized operating reserve dispatch and can only take values between 0 and 1. The budget uncertainty set can then be defined as follows:

$$W = \{w: (0 \leq w_t \leq 1\ \forall\, k = 1, \dots, t, \textstyle\sum_{k=1}^{t} w_k \leq \Gamma_t)\ \forall\, t \in T\} \tag{23}$$

where Γ_t is a so-called budget parameter that limits the cumulative operating reserve dispatch required up to time t and hence can be used to adjust the level of conservatism. In practice, dispatch of operating reserve is typically only requested a few times a year; hence, setting the budget parameter accordingly creates a more realistic uncertainty set and prevents the solution from being overly conservative.

4.1.2. Multistage affinely adjustable robust optimization

The traditional robust optimization approach is static, i.e. it does not account for recourse (reactive actions after the realization of the uncertainty). While the static approach is sometimes already sufficient, it may be too restrictive in other cases. Most plants that can provide operating reserve do have the flexibility to react to dispatch events and often have to do so in order to maintain feasible operations. In order to overcome the limitation of static robust optimization, the concept of adjustable robust optimization has been developed in recent years (Ben-Tal et al., 2004; Kuhn et al., 2011). The main idea is to include recourse in the form of decision rules that are functions of the uncertain parameters. Tractable formulations and solution methods can be developed if the decision rules are restricted to certain forms.

We apply the affinely adjustable robust optimization (AARO) approach in which the decision rules are affine functions of the uncertain parameters and can incorporate multistage decision-making. In the following, we briefly explain the general structure of the formulation. We consider the following multistage (n stages) optimization problem:

$$\min_x\{c^T x_1 : A_1(w)x_1 + \sum_{t=2}^{n} A_t x_t(w) \leq b \quad \forall\, w \in W\} \tag{24}$$

where x_t is the vector of the tth-stage decision variables. We assume that the objective function only depends on x_1 and that we have fixed recourse, i.e. only matrix A_1 is uncertain. While x_1 does not depend on w, x_t for $t \geq 2$ are recourse variables and hence do depend on the realization of w. Problem (24) is intractable since the set of possible functions for $x_t(w)$ is infinitely large. To obtain a tractable problem, we restrict ourselves to the set of affine functions and set $x_t(w) = \bar{x}_t + E_t w$, resulting in the following robust formulation by constraint-wise construction:

$$\min_{x_1,\bar{x},E}\left\{c^T x_1 : \max_{w \in W}\left\{a_{1,i}^T(w)x_1 + \sum_{t=2}^{n} a_{ti}^T(\bar{x}_t + E_t w)\right\} \leq b \quad \forall\, i\right\} \tag{25}$$

which can be reformulated into a single-level problem by applying standard robust counterpart techniques. In the robust counterpart formulation, the decision variables are x_1 as well as the parameters for the affine decision rules, $\bar{x}$ and E. In this way, AARO accounts for multistage recourse while maintaining computational tractability.

In the area of process systems engineering (PSE), AARO has first been applied by Zhang et al. (2016b) and Lappas and Gounaris (2016). Interestingly, there is a strong connection between adjustable robust optimization and flexibility analysis (Swaney and Grossmann, 1985), which was developed in the PSE community many years before robust optimization was established in the area of operations research. This relationship is discussed and theoretically shown for linear systems by Zhang et al. (2016c).

4.2. Case 1: Providing operating reserve with cryogenic energy storage

As mentioned before, the traditional way of providing operating reserve is by having power generating facilities on standby. Thus, some chemical plants with onsite generation capacities are capable of participating in the operating reserve market. In this case, we consider an air separation unit (ASU) with added cryogenic energy storage (CES) capability. The concept of CES is to store energy in the form of liquid gas and vaporize it when needed to drive a turbine for electricity generation. In an integrated ASU-CES system, as depicted in Fig. 19, electricity recovered from the CES can be used internally to power the ASU, to sell power back to the grid during higher-price hours, or to provide operating reserve.

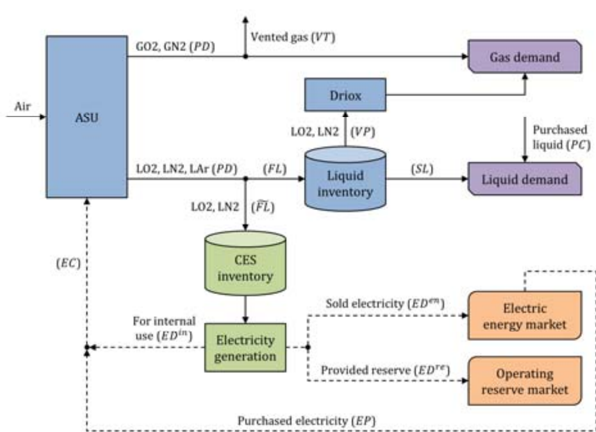


Fig. 19. Schematic of an integrated ASU-CES system (Zhang et al., 2015).

Apart from the production-related scheduling decisions, one has to decide how much of the liquid product is stored and how much electricity is generated for the difference purposes. In order to take the uncertainty in operating reserve dispatch into account, we use the budget uncertainty set presented in Section 4.1.1 and apply the traditional static robust optimization approach, as described in detail in Zhang et al. (2015).

We apply the proposed model to two instances with different levels of conservatism. The budget parameters are chosen such that in the first instance, maximum reserve dispatch can be requested up to three times over the entire scheduling horizon of one week, while in the second instance, it can be requested up to seven times a week. We compare the results for the two instances in Fig. 20, which shows the flows into and out of the CES tank as well as the resulting CES inventory profile for the scenario in which no operating reserve is dispatched.

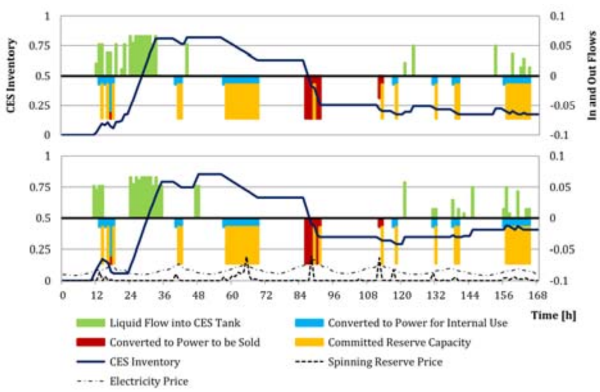


Fig. 20. Flows into and leaving the CES tank, and the resulting CES inventory for the scenario in which no operating reserve is dispatched for two instances with different levels of conservatism (Zhang et al., 2015).

In each time period in which operating reserve is provided, the CES inventory ensures that the maximum amount of operating reserve can be dispatched. As a result, the final CES inventory level depicted in Fig. 20 is not zero but has the value corresponding to the highest dispatch of operating reserve. Because of the larger budget parameters, the second instance arrives at a more conservative solution indicated by the higher final CES inventory. Nonetheless, we see that providing operating reserve with a CES system can be very attractive. Compared to the case where no operating reserve is provided, the cost savings achieved with operating reserve in the two instances are 7.0% and 2.8%, respectively.

4.3. Case 2: Providing operating reserve with interruptible load

Onsite power generation and energy storage require large investments and are not available at most chemical plants. However, even without power generation capabilities, industrial electricity consumers can provide operating reserve, namely in the form of interruptible load, which is often an even more attractive option. Interruptible load is defined as the amount of load reduction that the electricity consumer is willing to provide in case of contingency. The amount of interruptible load that can be provided depends strongly on the flexibility of the process and the ability of maintaining feasible operation when interruptible load is dispatched, i.e. when the plant has to be ramped down. Considering recourse decisions is crucial in this case; hence, we apply the AARO approach. The multistage adjustable robust model, which uses the discrete-time scheduling model developed by Zhang et al. (2016) as a basis, is presented in detail in Zhang et al. (2016b).

In the proposed model, the first-stage decisions are the ones related to the base operation (no load reduction required during the entire scheduling horizon) of the plant as well as how much interruptible load to provide. The recourse decisions are production and product purchase deviations from the base operation. The decision rule for the actual production rate in time period t, for instance, is defined as follows:

$$P_t(w) = \bar{P}_t + \sum_{k=t-\zeta_t}^{t} p_{tk}\, w_k \tag{26}$$

where $\bar{P}_t$ is the target production rate, and p_{tk} are variables that define the affine decision rule. This is a multistage formulation since at each time period t, the recourse decision depends on the uncertain parameters that have been realized in the preceding ζ_t time periods as well as the current time period, i.e. w_k for $t - \zeta_t \leq k \leq t$. The possible extent of recourse decisions, and with that the level of conservatism, can be varied by adjusting ζ_t.

We introduce an auxiliary parameter $\bar{\zeta}$, which denotes the maximum number of previously realized uncertain parameters that are considered in the decision rules, and set ζ_t such that $\zeta_t = \min\{\bar{\zeta}, t-1\}$. If $\bar{\zeta} = 0$, only the uncertain parameter from the current time period appears in the decision rules; hence, the only possible recourse actions are the reduction in production rate when load reduction is requested and additional product purchase in the same time periods. If $\bar{\zeta} > 0$, i.e. uncertain parameters from previous time periods are also taken into account, lost production can also be made up by increasing production or purchase in time periods after the load reduction occurred.

Applying the proposed model to an industrial air separation case study, for which the data are provided by Praxair, the scheduling problem is solved for a one-week time horizon with an hourly time discretization. The uncertainty set is chosen such that request for maximum load reduction can occur up to seven times a week. The solution strongly depends on the extent of recourse that is considered in the model. With $\bar{\zeta} = 0$, a cost reduction of 1.2% is achieved compared to the case in which no interruptible load is provided. These cost savings further increase by more than 50% if $\bar{\zeta}$ is changed to 23. However, this improvement in the quality of the solution comes at the cost of deteriorating computational performance. In the case of $\bar{\zeta} = 0$, the model has 3,282 binary variables, 82,670 continuous variables, and 84,604 constraints, while for $\bar{\zeta} = 23$, the number of binary variables remains the same, but the numbers of continuous variables and constraints increase to 330,242 and 325,000, respectively. The computation times required to solve the models to 0.1% optimality gap using CPLEX 12.6 on an Intel Core i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM are 185 s and 6,476 s. The parameter $\bar{\zeta}$ can be further increased (up to 167). However, computational experiments show only marginal improvement in the solution for $\bar{\zeta} > 23$; hence $\bar{\zeta} = 23$ is chosen as a good trade-off between level of conservatism and problem size.

The results for the case with greater extent of recourse, i.e. $\bar{\zeta} = 23$, are shown in Fig. 21 and Fig. 22. Along with the electricity and interruptible load prices, Fig. 21 shows the target load profile for the plant as well as the amount of interruptible load provided, which has to be less than the target electricity consumption. For liquid oxygen (LO2), one of the products, Fig. 22 shows the inventory profile and the corresponding product flows as well as the cumulative recourse actions in terms of changes in production and purchase rates. Negative production recourse indicates time periods in which interruptible load is provided. One can see that the vast majority of the lost production is made up by increasing production after load reduction (positive production recourse).

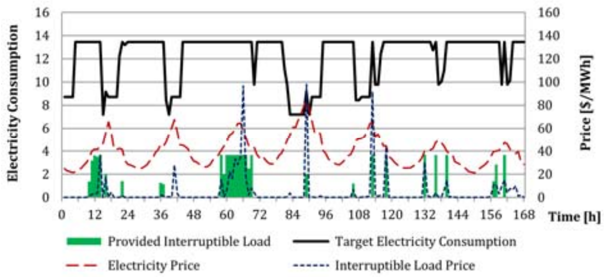


Fig. 21. Target electricity consumption profile and provided interruptible load for the case of $\bar{\zeta} = \mathbf{23}$, and price profiles (Zhang et al, 2016b).

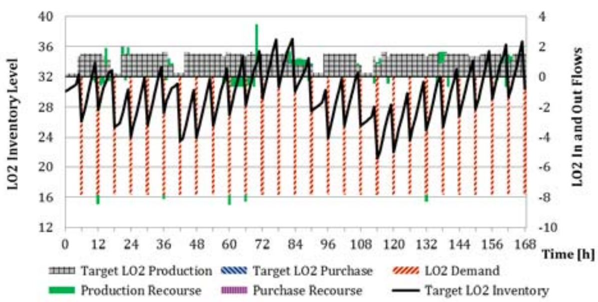


Fig. 22. Target and recourse LO_2 flows and target inventory profile (Zhang et al, 2016b).

Furthermore, it is worth mentioning that contrary to results in the literature indicating that demand response is more effective in plants with lower utilization, we find that this is not true when interruptible load is provided. Here, the largest cost savings are achieved at a high, yet not maximum level of plant utilization. The explanation is that higher plant utilization allows larger amount of interruptible load to be provided, yet some flexibility is still required for the implementation of effective recourse.

5. Integration of planning and scheduling

5.1. Background

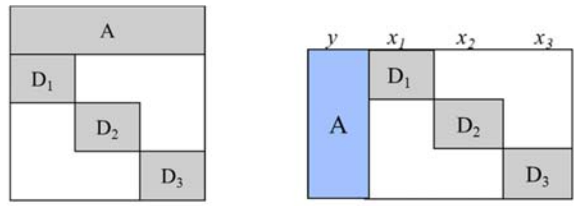
Reviews on production planning and scheduling in the chemical, petrochemical, and pharmaceutical industries can be found in Shah (2004), Sung and Maravelias (2007), Maravelias and Sung (2009), Verderame et al. (2010), Li and Ierapetritou (2009) and Shah et al. (2011). Planning formulations tend to be Linear Programming (LP), or sometimes Mixed-Integer Linear or Nonlinear Programming (MILP/MINLP), models. Constraints mainly include material and inventory balances that account for raw material purchases, production amounts, and sales. Decision variables typically include production rates, inventory levels, and raw material, intermediate, and finished product flows. The objective function is usually a financial performance indicator, such as profit or total costs, which include revenue from sales, and operating, transportation, and inventory costs Kallrath (2002).

For scheduling, the objective function may correspond to minimizing makespan or tardiness. Note that in general the planning model is not concerned with the sequencing of operations in each plant or reactor. However, when applied to multi-product plants, the lack of a rigorous treatment of the production sequencing may underestimate the total costs and yield a plan that is not feasible. Without adding considerable complexity to the planning model, Erdirik-Dogan and Grossmann (2007) proposed a Detailed Planning (DP) model that utilizes Traveling Salesman Problem (TSP) constraints to generate a cyclic schedule, which is broken in one link to yield an optimal sequence with minimum changeover times. The formulation allows sequence-dependent changeovers across time periods. The planning model used in this work is based on the DP model. Liu et al. (2008) have also investigated the estimation of the sequencing using TSP constraints. Sung and Maravelias (2009) proposed an algorithm for identifying the projection of a scheduling model's feasible region onto the space of production targets. The projected feasible region, which is generated with the program Quickhall, can then be used to address integrated production planning and scheduling problems. These authors illustrated their methodology with the State-Task Network model (Kondili et al, 1993).

At the scheduling level, a number of models that use either discrete- or continuous-time representation of events have been proposed (Floudas and Lin, 2004). In this section, we will focus on continuous-time models. For sequential batch/continuous processes, two event representations have received a great deal of attention: time slots and precedence based.

The main idea in the use of time slots is that each product can be assigned to a specific slot that has variable length. In some cases, especially in continuous processes, the number of time slots to be allocated for each reactor is known a priori. However, in batch processes that may not be true and additional time slots have to be allocated, thus increasing the size of the problem (Erdirik- Dogan & Grossmann, 2008). Likewise in the aforementioned DP formulation, the works by Lima, Grossmann, & Jiao (2011) and Kopanos et al.(2011) have the desirable feature of sequence-dependent changeovers across adjacent time periods, which even though adds more complexity to the model due to the larger number of binary and continuous variables, renders more realistic plans by not imposing a "hard" barrier for changeovers across time periods.

Unlike time slot models, precedence-based scheduling models effectively model changeovers by the use of disjunctions that are commonly reformulated as Big-M constraints. Four types of precedence-based models have been proposed: Unit-Specific Immediate Precedence (USIP) (Cerdá et al., 1997), Immediate Precedence (IP) (Méndez et al., 2000), General Precedence (GP) (Méndez et al., 2001), and Unit-Specific General Precedence (USGP) (Kopanos et al., 2009). Briefly, IP and USIP differ in that the latter only takes into account the immediate precedence of products that are assigned to the same processing unit, whereas the former does not. The GP model generalizes the concept of precedence by accounting for all precedence relations (immediate or not) and requires fewer binary variables than the immediate precedence models, but cannot account for changeovers costs. To overcome that limitation, the USGP model was proposed.



$$\begin{aligned} &\max\ c^T x \\ &st\ \ Ax = b \\ &\quad D_i x_i = d_i\ \ i = 1,..n \\ &\quad x \in X = \{x | x_i\ , i = 1,..n, | x_i \geq 0\} \end{aligned}$$

$$\begin{aligned} &\max\ a^T y + \sum_{i=1,..n} c_i^T x_i \\ &st\ \ Ay + D_i x_i = d_i\ \ i = 1,..n \\ &\quad y \geq 0,\ x_i \geq 0, i = 1,..n \end{aligned}$$

Fig. 23. Schematic of the set of constraints of a decomposable optimization problem with complicating constraints (on the left) and variables (on the right).

When attempting to solve large-scale industrial problems, some authors have identified the need to decompose the problem into subproblems (Conejo et al., 2006). Decomposition approaches take advantage of specific structures of the optimization problems. Two such cases arise in practice: the complicating constraint and the complicating variable structures, as shown in Fig. 23. Note that complicating constraints involve variables from different blocks (e.g. inventory), and complicating variables link constraints pertaining to different blocks (e.g. stage 1 in stochastic programming).

Different decomposition approaches have been proposed depending on the nature of the optimization problem (e.g., linear vs. nonlinear, purely continuous vs. mixed-integer). For LP and MILP problems with complicating constraints (Fig. 23a), Lagrangean decomposition (Guignard, 2003) and Branch-and-Price (Barnhart et al., 1998) have been applied. When complicating variables are present (Fig. 23b), Benders decomposition is typically used (Benders, 1962) although it is restricted to having only integer variables in the first-stage decisions for stochastic programming. It should be noted that problems with complicating variables can always be converted to problems with complicating constraints, which are then solved by Lagrangean Decomposition (a particular case of Lagrangean Relaxation (Guignard, 2003)). The main idea of such approach is to create copies of a subset of the variables (called copy variables) and add equality constraints that equate the original and the copy variables. This yields a problem with complicating constraints, which are dualized to allow decomposition. Another decomposition approach is the Bilevel Decomposition proposed by Iyer and Grossmann (1998), which is similar to Benders decomposition except that it relies on tailor made MILP problems that yield valid bounds. This approach involves solving an aggregate formulation in the upper level and a detailed formulation in the lower level with some decisions fixed by the upper level (e.g. see section 3.3.4).

5.2. Major approaches to integration of planning and scheduling

A review of methods and decomposition approaches for the integration of planning and scheduling can be found in Maravelias and Sung (2009). The three major approaches that have emerged for integrating planning and scheduling decisions are the following:

1. Relaxation/Aggregation of detailed scheduling model

2. Projection of scheduling model onto Planning level decisions
3. Iterative decomposition of Planning and Scheduling Models

Representative of the first approach is the work by Erdirik-Dogan & Grossmann (2007, 2008). For the second approach, it is the work by Sung and Maravelias (2009). For the third, it is Terrazas and Grossman (2011a) and Calfa et al. (2013). In the next sections, we illustrate applications with the first and third approaches since they deal with sequence dependent changeovers that are a major issue in multistage plants.

5.3. Relaxation/aggregation of detailed scheduling model

Erdirik-Dogan and Grossmann (2007, 2008) addressed the planning and scheduling of multistage batch and continuous processes with sequence dependent changeovers. The authors considered an MILP scheduling model based on continuous time domain representation, based on time slots. The model, which involves mass balances and intermediate storage, handles rigorously sequence dependent changeover times. One obvious way to integrate planning and scheduling is to use the detailed MILP scheduling over the entire planning horizon. The drawback with such an approach is that the size of the MILP can become unmanageable and virtually impossible to solve. Therefore, these authors considered two major options for replacing the detailed timing constraints of the MILP: a) develop a relaxed planning model by eliminating the detailed timing constraints and replacing them with simplified constraints that underestimate the sequence dependent changeover times. b) develop a more detailed planning model by adding sequencing constraints that rigorously account for transitions. While option (a) leads to a simpler MILP planning model it will tend to overestimate the profit due to the oversimplification of the scheduling model. In contrast option (b) leads to a somewhat more complex MILP which, however, provides much more accurate estimates of the profit as it anticipates more accurately the effects of the scheduling level.

The basic idea in the use of the traveling salesman (TSP) constraints is as follows (see Fig. 24). First, for given changeover times $\tau_{ii'}$ between two successive products i, i', a cyclic schedule of different products is generated where the total transition or changeover time is minimized using the binary variables $ZP_{ii'mt}$,. Next, the cycle is broken at a pair in order to obtain a sequence with the binary variable $ZZP_{ii'mt}$.

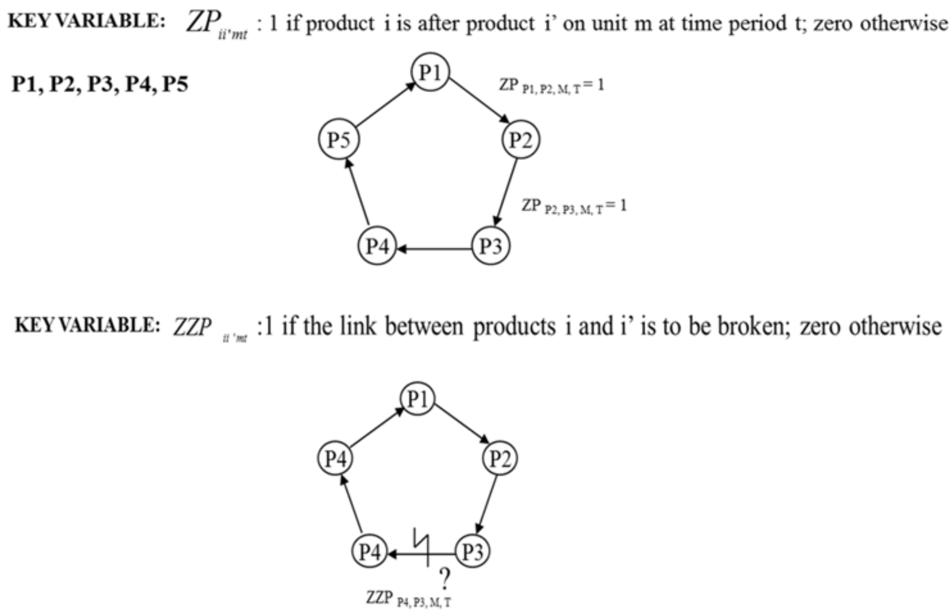


Fig. 24. Use of 0-1 variables to determine sequence ($ZP_{ii'mt}$) and to break the cycle ($ZZP_{ii'mt}$).

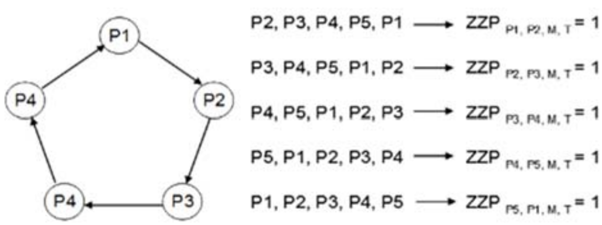


Fig. 25. Different sequences obtained from corresponding broken link ($ZZP_{ii'mt}$=1).

As shown in Fig. 25, the sequence obtained will depend on which link is broken.

The above sequencing is based on the traveling salesman problem (TSP) and can be accomplished with the following set of integer constraints TSP (see Erdirik-Dogan and Grossmann, 2007):

$$
\begin{aligned}
& YP_{imt} = \sum_{i'} ZP_{ii'mt} \;\; \forall i, m, t \\
& YP_{i'mt} = \sum_{i} ZP_{ii'mt} \;\; \forall i', m, t \\
& YP_{imt} \wedge [\wedge_{i' \neq i} (\neg YP_{i'mt})] \Leftrightarrow ZP_{iimt} \; \forall i, m, t \\
& YP_{imt} \geq ZP_{iimt} \; \forall i, m, t \\
& ZP_{iimt} + YP_{i'mt} \leq 1 \; \forall i, i' \neq i, m, t \\
& ZP_{iimt} \geq YP_{imt} - \sum_{i' \neq i} YP_{i'mt} \;\; \forall i, m, t \\
& \sum_i \sum_{i'} ZZP_{ii'mt} = 1 \; \forall m, t \\
& ZZP_{ii'mt} \leq ZP_{ii'mt} \; \forall i, i', m, t
\end{aligned}
\qquad \text{(TSP)}
$$

where the sequencing variables $ZP_{ii'mt}$ are included in the objective function in the form of fixed costs. Having determined the sequence, the total transition or changeover time within each week can be determined with Eq. (27) (see Fig. 26):

$$TRNP_{mt} = \sum_i \sum_{i'} \tau_{ii'} ZP_{ii'mt} - \sum_i \sum_{i'} \tau_{ii'} ZZP_{ii'mt} \ \forall m, t \tag{27}$$

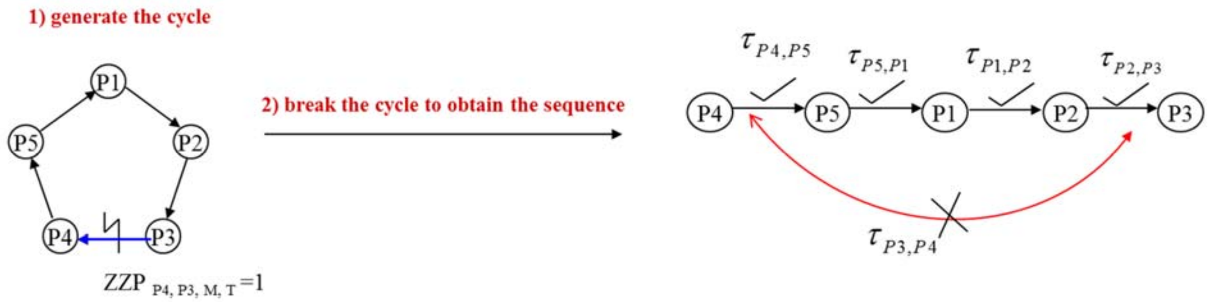


Fig. 26. Determining total changeover times from a given sequence.

Edririk-Dogan and Grossmann (2007, 2008) applied this approach to the planning and scheduling of multistage batch plants with excellent results in that often the profit predicted by the relaxed planning model with the TSP constraints gave the same profit as if the detailed scheduling model was applied to the given planning horizon. In the following section, we present an example for multiperiod refinery planning in which the TSP constraints are applied.

5.3.1. Multiperiod refining planning

We assume we are given a refinery configuration (e.g. Fig. 27), a specified time horizon with N time periods, and inventories and sequence-dependent changeover times of M crudes. The problem is to determine what crude oils to process and in which time period, the amounts of these crude oils to process, and the sequence of processing these crudes. 0-1 variables are used to assign crude in period t, to indicate position of crude in sequence, and to indicate where the cycle is broken. Continuous variables are used for flows, inventories and cut temperatures in the Fractionation Index model (see Alattas et al., 2012). Qualitatively, the MINLP optimization model is as follows:

$$\max Profit = Product\ sales\ minus\ the\ costs\ of\ product\ inventory, crude\ oil,$$
$$unit\ operation\ and\ net\ transition\ times$$
$$\text{s.t.}\ Performance\ CDU\ (Fractionation\ Index\ Model)\ each\ crude, each\ time\ period \qquad \text{(MINLP)}$$
$$Mass\ balances, inventories\ each\ crude, each\ time\ period$$
$$Sequencing\ constraints\ (TSP)$$

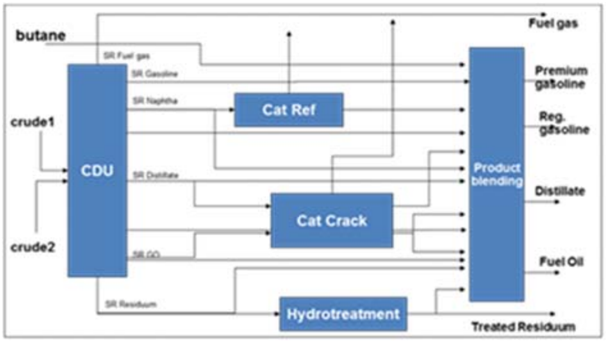


Fig. 27. Petroleum refinery processing several crudes.

This model, which incorporates a nonlinear model for the CDU (fractionation index), was applied to a refinery that processes 5 crudes over a time horizon of 4 weeks. The products include fuel gas, regular gasoline, premium gasoline, distillate, fuel oil and treated residue. The MINLP model, which involved 13,680 variables (900 0-1) and 15,047 constraints was solved with DICOPT (CPLEX, CONOPT) in only 37 seconds. The optimal schedule, which is shown in Fig. 28, yielded a profit of $23,994,000. The breakdown of the costs is given in Table 7. This example then shows how sequence dependent changeover times in refinery operations can be effectively handled with the constraints (TSP).

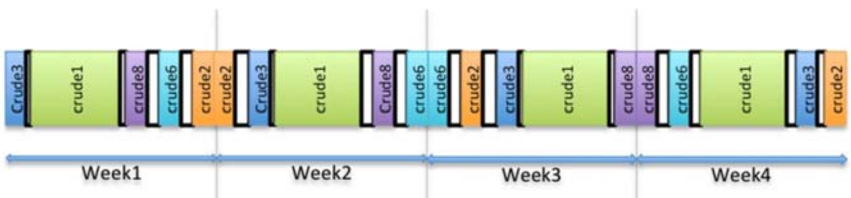


Fig. 28. Optimal schedule for multiperiod refinery planning.

Table 7. Economics of optimal refinery planning ($1,000).

Profit	23,994
Sales	223,684
Crude oil cost	162,988
Other feedstock	446
Inventory cost	1,256
Operating cost	32,510
Changeover cost	2,480

5.4. Iterative decomposition of planning and scheduling models

In this section, we address the integration of multisite planning and scheduling in which the multiscale optimization challenge involves spatial and temporal dimensions. We consider specifically the multi-period integrated planning and scheduling of a network of multiproduct batch plants located in multiple sites addressed by Calfa et al. (2013) (see Fig. 29).

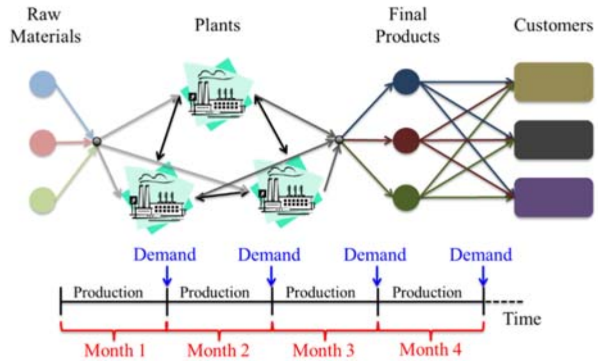


Fig. 29. Supply chain for multisite multiproduct facilities.

In the network of Fig. 29, a set of raw materials is purchased and transported to each plant, which transforms them into products to fulfill customer demands that are specified over several months. Shipments of intermediate products between plants are allowed. Products are classified into groups or families. Sequence-dependent changeover data are given between groups of products, which are considered at the planning level. For the scheduling level, changeover times and costs associated with the transition from one product to another of the same group are also given. The sequence-dependent changeovers between different groups of products are significantly higher than the ones from one product to another in the same group. In addition, the assignment of a product to a plant that does not normally produce it incurs a set-up cost. The goal is to minimize the total cost over the time horizon given by time periods of months in which the demands are specified at the end of each period.

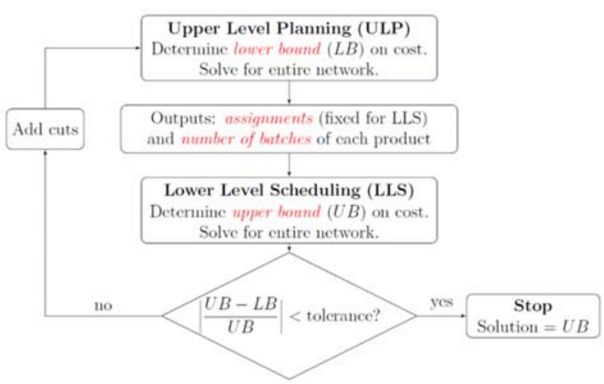


Fig. 30. Bi-level decomposition for integrating planning and scheduling.

To accomplish the integration of planning and scheduling we decompose the problem into two major levels as shown in Fig. 30. The upper level corresponds to the planning model over the given time horizon, a simplified MILP, in which the impact of changeovers is captured through the constraints (TSP) (see Calfa et al., 2013). This model yields a lower bound on the total cost, and determines the assignment of products to each plant and their corresponding number of batches for each time period. The Lower Level Scheduling (LLS), which yields an upper bound, is concerned with the scheduling of each of the plants with their corresponding MILP model in which the changeovers are rigorously accounted for (see Calfa et al., 2013).

While at the lower level, the MILP scheduling model for each plant can be solved independently, the MILP for the Upper Level Planning (ULP) problem can become expensive to solve as it has to optimize the entire network simultaneously. In order to address this difficulty, a temporal Lagrangean decomposition is applied (Terrazas et al, 2011b) at the Upper Level Planning as shown in Fig. 31 and Fig. 32.

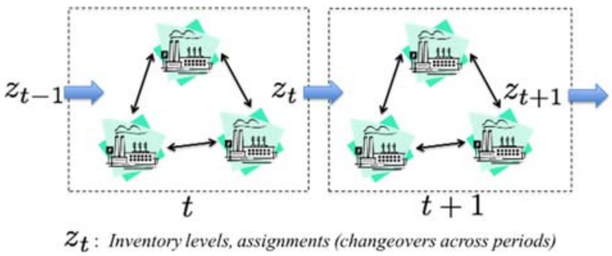


Fig. 31. Temporal Lagrangean decomposition (TLD).

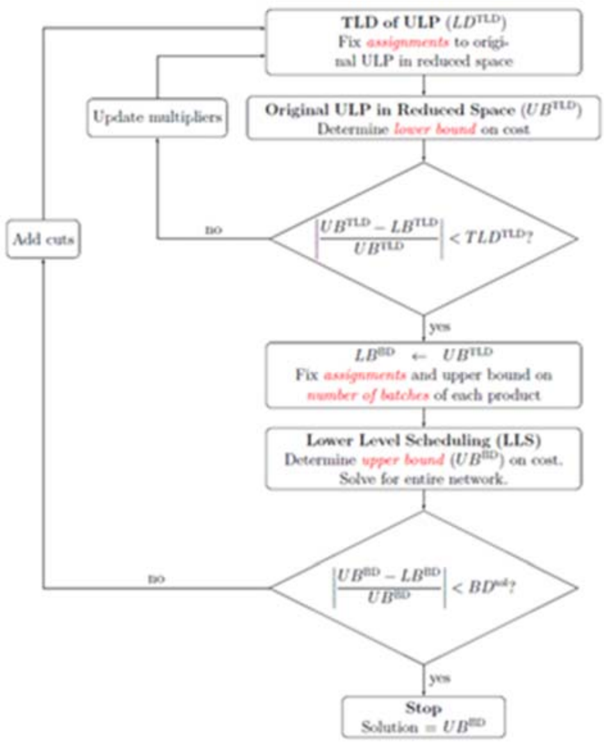


Fig. 32. Integration of temporal Lagrangean decomposition in bi-level decomposition.

We should note that the temporal Lagrangean subproblems are solved in parallel, the multipliers are updated with the subgradient method, and a maximum number of 30 iteration is allowed if convergence of the bounds for the Upper Level Planning is not achieved.

The network consists of 2 plants in which plant 1 involves units U11 and U12, while plant 2 involves unit U21. There are 3 raw materials and 3 customers. Fig. 33 presents the schedule of week 1 of a 4-week time horizon. As it can be seen for the first week, plant 1 is assigned the production of products C, D and B, while plant 2 is assigned products A, B and E. The numbers in parenthesis correspond to the number of batches.

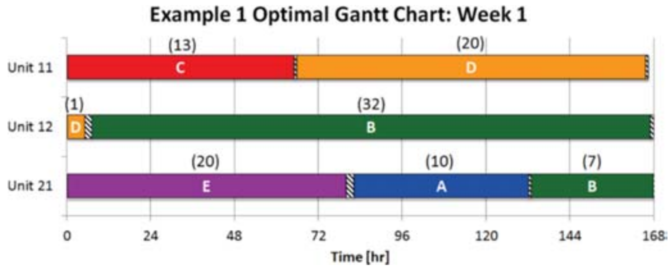


Fig. 33. Gant chart for first week in 2 plant network.

Table 8 presents the results for examples of increasing size in order to compare the hybrid bi-level/Lagrangean decomposition technique with the direct solution of the full space (FS) MILP model using GAMS/Gurobi 4.6. Example 1 has 2 plants/3 units over 4 weeks, example 2 has 3 plants/6 units over 6 weeks, and example 3 has 5 plants/10 units over 12 weeks. It is clear that the hybrid decomposition scheme is computationally much more effective than solving the full-space MILP scheduling model as problem size increases.

Table 8. Computational results for problems of increasing size.

	Ex.	Problem	Discrete Variables	Continuous Variables	Constraints	NZ Elements.	Nodes	Time (s)
4 weeks	1	ULP	528	925	1,412	4,537	5,015	0.99
		LLS	507	1,039	1,726	5,049	29	0.18
		FS	936	1,201	2,924	9,113	94,929	44.98
6 weeks	2	ULP	6,328	52,783	43,169	145,009	57	2.34
		LLS	4,412	53,047	45,378	145,831	0	1.62
		FS	128,400	95,563	437,649	3,998,885	57,536	12,229
12 weeks	3	ULP	119,397	834,195	590,810	2,206,546	0	4,070
		LLS	228,701	898,119	1,140,007	6,836,510	0	452.5
		FS	6,726,779	3,138,985	22,895,121	648,785,966	-	-

6. Concluding remarks

In this paper, we have shown that for scheduling, a variety of powerful approaches are available, such as the STN & RTN discrete/continuous-time models, which have reached maturity. We have shown that GDP facilitates formulation of complex constraints, widening the modeling scope of these problems. Through application such as blending, we have shown the increased emphasis on the global optimization of nonlinear models (MINLP).

In the area of Demand Side Management, we have shown that the link with electric power yields a new area of application of scheduling. We have shown that large-scale MILP models that involve detailed operational decisions can yield significant economic savings such as in the case of cryogenic energy storage and interruptible load, in which uncertainty in the operating reserve dispatch was handled through adjustable robust optimization.

In the area of Integration of Planning and Scheduling we have shown that this is still a largely unsolved problem in which no single approach has emerged as a winner. We focused, however, on the handling of changeovers at the planning level with traveling salesman constraints in order to capture the impact of changeovers at the detailed scheduling operations. We also showed the need for decomposition techniques such as Bi-level and Lagrangean decomposition. Applications were presented for optimizing multiperiod planning of refineries and multisite planning and scheduling of multiproduct batch facilities.

Finally, it is clear that a number of major challenges remain in the scheduling area. These include:

a) The modeling challenge: Integration of planning, scheduling, and control models for the various components of the supply chain, including nonlinear process models.
b) The multiscale optimization challenge: Coordinated optimization of planning and scheduling models over geographically distributed sites, and over the long-term (years), medium-term (months) and short-term (days, min) decisions.
c) The uncertainty challenge: Anticipating impact of uncertainties in a meaningful way in both longer term planning problem and shorter term scheduling problems.
d) Algorithmic and computational challenges: Effectively solving large-scale MILP/MINLP models including nonconvex problems in terms of efficient algorithms, and modern computer architectures.

Acknowledgment

Financial support from Fundação para a Ciência e Tecnologia through projects IF/00781/2013, UID/MAT/04561/2013, and the Center of Advanced Process Decision-making at Carnegie Mellon. Thanks also to Danielle Zyngier, Optimization Specialist from Hatch, for providing the fruit juice processing example used to illustrate UOPSS.

References

Adonyi, R., Romero, J., Puigjaner, L., Friedler, F., 2003. Incorporating Heat Integration in Batch Process Scheduling. Applied Thermal Engineering 23, 1743-1762.

Alattas, A., Grossmann, I.E., Palou Rivera, I., 2012. A Multiperiod MINLP Model for Refinery Production Planning. Ind. Eng. Chem. Res. 51, 12852–12861.

Alfaki, M., Haugland, D., 2013. Strong formulations for the pooling problem. J. Global Optim. 56, 897-916.

Balas, E., 1979. Disjunctive programming. Annals of Discrete Mathematics 5, 3-51.

Balas, E., 1985. Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. SIAM J. Algebraic Discrete Methods 6, 466-486.

Baldea, M., Harjunkoski, I., 2014. Integrated production scheduling and process control: a systematic review. Comp. Chem. Eng. 71, 377-390.

Barbosa-Póvoa, A., Macchietto, S., 1994. Detailed Design of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 18, 1013-1042.

Barnhart, C.; Johnson, E.L.; Nemhauser, G. L.; Savelsbergh, M.W. P.; Vance, P.H. 1998. Branch-and-price: column generation for solving huge integer programs, Operations Research, 46 (3): 316–329

Benders, J. F. 1962. Partitioning procedures for solving mixed-variables programming problems, Numeri. Math. vol. 4, pp. 238–252 (1962).

Ben-Tal, A., Eiger, G. Gershovitz, V., 1994. Global minimization by reducing the duality gap. Math Programming 63, 193-212.

Ben-Tal, A., Goryashko, A., Guslitzer, E., Nemirovski, A, 2004. Adjustable robust solutions of uncertain linear programs. Mathematical Programming, 99(2), 351–376.

Ben-Tal, A., El Ghaoui, L., Nemirovski, A., 2009. Robust Optimization. Princeton University Press.

Bergamini, M.L, Aguirre, P., Grossmann, I.E., 2005. Logic-based outer approximation for globally optimal synthesis of process networks. Comp. Chem. Eng. 29, 1914-1933.

Bertsimas, D., Sim, M., 2004. The Price of Robustness. Operations Research, 52(1), 35–53.

Biegler, L.T., Grossmann, I.E., Westerberg, A.W., 1997. Systematic Methods of Chemical Processing Design. Prentice Hall: Upper Saddle River, N, USA.

Blomer, F., Gunther, H.O., 2000. LP-based heuristics for scheduling chemical batch processes. Int. J. Prod. Res. 38, 1029-1051.

Boland, N., Kalinowski, T., Rigterink, F., 2016. New multi-commodity flow formulations for the pooling problem. J. Global Optim. 66, 669-710.

Cafaro, D.C., Cerdá, J., 2004. Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation. Comput. Chem. Eng. 28, 2053-2068.

Cafaro, D.C., Cerdá, J., 2011. A rigorous mathematical formulation for the scheduling of tree-structure pipeline networks. Ind. Eng. Chem. Res. 50, 5064-5085.

Cafaro, D.C., Cerdá, J., 2012. Rigorous scheduling of mesh-structure refined petroleum pipeline networks. Comput. Chem. Eng. 38, 185-203.

Cafaro, D.C., Cerdá, J., 2014. Rigorous formulation for the scheduling of reversible-flow multiproduct pipelines. Comput. Chem. Eng. 61, 59-76.

Calfa, B.A., Agarwal, A., Grossmann, I.E., Wassick, J.M., 2013. Hybrid Bilevel-Lagrangean Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants. Ind. Eng. Chem. Res. 52, 2152−2167.

Castillo-Castillo, P.A., Mahalec, V., 2016. Improved continuous-time model for gasoline blend scheduling. Comp. Chem. Eng. 84, 627-646.

Castro, P.M., 2010. Optimal scheduling of pipeline systems with a resource-task network continuous-time formulation. Ind. Eng. Chem. Res. 49, 11491-11505.

Castro, P.M., 2015. New MINLP Formulation for the Multiperiod Pooling Problem. AIChE J. 61, 3728-3738.

Castro, P.M., 2015b. Tightening piecewise McCormick relaxations for bilinear problems. Comp. Chem. Eng. 72, 300-311.

Castro, P.M., 2016. Source-based discrete and continuous-time formulations for the crude oil pooling problem. Comp. Chem. Eng. 93, 382-401.

Castro, P.M., 2016b. Improving energy efficiency in batch plants through direct heat integration. In Alternative Energy Sources and Technologies: Process Design and Operation, Mariano Martin (ed.), 341-362. Springer.

Castro, P.M., 2017. Spatial branch-and-bound algorithm for MIQCPs featuring multiparametric disaggregation. Optimization Methods & Software 32, 719-737.

Castro, P.M., 2017b. Optimal Scheduling of Multiproduct Pipelines in Networks with Reversible Flow. Ind. Eng. Chem. Res. Under review.

Castro, P., Barbosa-Póvoa, A.P.F.D., Matos, H., 2001. An Improved RTN Continuous-time Formulation for the Short-term Scheduling of Multipurpose Batch Plants. Ind. Eng. Chem. Res. 40, 2059-2068.

Castro, P.M., Barbosa-Póvoa, A.P., Matos, H.A., Novais, A.Q., 2004. Simple Continuous-Time Formulation for Short-Term Scheduling of Batch and Continuous Processes. Ind. Eng. Chem. Res. 43, 105-118.

Castro, P.M., Custódio, B., Matos, H.A., 2015. Optimal scheduling of single stage batch plants with direct heat integration. Comp. Chem. Eng. 82, 172-185.

Castro, P.M., Grossmann, I.E., 2005. New continuous-time MILP model for the short-term scheduling of multistage batch plants. Ind. Eng. Chem. Res. 44, 9175-9190.

Castro, P.M., Grossmann, I.E., Novais, A.Q., 2006. Two new continuous-time models for the scheduling of multistage batch plants with sequence dependent changeovers. Ind. Eng. Chem. Res. 45, 6210-6226.

Castro, P.M., Grossmann, I.E., 2012. Generalized Disjunctive Programming as a Systematic Modeling Framework to Derive Scheduling Formulations. Ind. Eng. Chem. Res. 51, 5781-5792.

Castro, P.M., Grossmann, I.E., 2014. Global Optimal Scheduling of Crude Oil Blending Operations with RTN Continuous-time and Multiparametric Disaggregation. Ind. Eng. Chem. Res. 53, 15127-15145.

Castro, P.M., Mostafaei, H., 2017. Product-centric continuous-time formulation for pipeline scheduling. Comp. Chem. Eng. 104, 283-295.

Castro, P.M., Rodrigues, D., Matos, H.A., 2014. Cyclic scheduling of pulp digesters with integrated heating tasks. Ind. Eng. Chem. Res. 53, 17098-17111.

Castro, P.M., Sun, L., Harjunkoski, I., 2013. Resource−Task Network Formulations for Industrial Demand Side Management of a Steel Plan. Ind. Eng. Chem. Res. 52, 13046-13058.

Cerdá, J., Henning, G. P., Grossmann, I. E., 1997. A Mixed-Integer Linear Programming Model for Short-Term Scheduling of Single-Stage Multiproduct Batch Plants with Parallel Lines. Ind. Eng. Chem. Res. 36, 1695–1707.

Chen, C.L., Chang, C.-Y., 2009. A resource-task network approach for optimal short-term/periodic scheduling and heat integration in multipurpose batch plants. App. Therm. Eng. 29, 1195-1208.

Clocksin, W.F., Mellish, C.S., 1981. Programming in Prolog. New York: Springer-Verlag.

Conejo, A. J., Castillo, E., Mínguez, R., García-Bertrand, R., 2006. Decomposition Techniques in Mathematical Programming: Engineering and Science Applications. Springer Science+Business Media. New York, NY. USA.

Dias, L.S., Ierapetritou, M., 2016. Integration of scheduling and control under uncertainties: review and challenges. Chem. Eng. Res. Des. 116, 98-113.

Erdirik-Dogan, M., Grossmann, I.E., 2007. Planning Models for Parallel Batch Reactors with Sequence-Dependent Changeovers. AIChE Journal. 53, 2284–2300.

Erdirik-Dogan, M., Grossmann, I.E., 2008. Slot-Based Formulation for the Short-Term Scheduling of Multistage, Multiproduct Batch Plants with Sequence-Dependent Changeovers. Ind. Eng. Chem. Res. 47, 1159–1183.

Fernández, I., Renedo, C.J., Pérez, S.F., Ortiz, A., Mañana, M., 2012. A review: Energy recovery in batch processes. Renewable and Sustainable Energy Reviews 16, 2260-2277.

Floudas, C.A., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Comp. Chem. Eng. 28, 2109-2129.

Glismann, K., Gruhn, G., 2001. Short-term scheduling and recipe optimization of blending processes. Comp. Chem. Eng. 25, 627-634.

Grossmann, I.E., 2005. Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. AIChE J. 51, 1846-1857.

Grossmann, I.E., 2012. Advances in Mathematical Programming Models for Enterprise-Wide Optimization. Comp. Chem. Eng. 47, 2-18.

Grossmann, I.E, Trespalacios, F., 2013. Systematic Modeling of Discrete-Continuous Optimization Models through Generalized Disjunctive Programming. AIChE J. 59, 3276-3295.

Guignard, M., 2003. Lagrangean Relaxation. Top. 11, 151–228.

Gupta, S., Karimi, I.A., 2003. An improved MILP formulation for scheduling multiproduct, multistage batch plants. Ind. Eng. Chem. Res. 42, 2365–2380.

Halim. I., Srinivasan, R., 2009. Sequential Methodology for Scheduling of Heat-Integrated Batch Plants. Ind. Eng. Chem. Res. 48, 8551-8565.

Hane, C.A., Ratliff, H.D., 1995. Sequencing inputs to multi-commodity pipelines. Annals of Operations Research 57, 73-101.

Harjunkoski, I., Grossmann, I., 2002. Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. Comp. Chem. Eng. 26, 1533–1552.

Harjunkoski, I., Maravelias, C., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J., 2014 Scope for Industrial Applications of Production Scheduling Models and Solution Methods. Comput. Chem. Eng. 62, 161-193.

Haverly, C.A., 1978. Studies of the behavior of recursion for the pooling problem. SGMAP Bulletin 25, 19-28.

Herran, A., Cruz, J.M., Andres, B., 2010. Mathematical model for planning transportation of multiple petroleum products in a multi-pipeline system. Comput. Chem. Eng. 34, 401-413.

Ierapetritou, M.G., Floudas, C.A., 1998a. Effective continuous-time formulation for short-term scheduling: 1. Multipurpose batch processes. Ind. Eng. Chem. Res. 37, 4341–4359.

Ierapetritou, M.G., Floudas, C.A., 1998b. Effective continuous-time formulation for short-term scheduling: 2. Continuous and semi-continuous processes. Ind. Eng. Chem. Res, 37, 4360-4374.

Iyer, R.R., Grossmann, I.E., 1998. A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks. Ind. Eng. Chem. Res. 37, 474–481.

Jeroslow, R.G., Lowe, J.K., 1984. Modelling with integer variables. Math. Programming Studies 22, North-Holland, Amsterdam, pp 167-184.

Jia Z., Ierapetritou, M., Kelly, J.D., 2003. Refinery short-term scheduling using continuous time formulation: crude-oil operations. Ind. Eng. Chem. Res. 42, 3085-3097.

Kallrath, J., 2002. Planning and scheduling in the process industry. OR Spectrum 24, 219-250.

Karuppiah, R., Grossmann, I.E., 2006. Global optimization for the synthesis of integrated water systems in chemical processes. Comp. Chem. Eng. 30, 650-673.

Kelly, J.D., 2003. Next Generation Refinery Scheduling Technology. NPRA Plant Automation and Decision Support Conference, CC-03-153.

Kolodziej, S., Castro, P.M., Grossmann, I.E., 2013b. Global optimization of bilinear programs with a multiparametric disaggregation technique. J. Glob. Optim. 57, 1039-1063.

Kolodziej, S.P., Grossmann, I.E., Furman, K.C., Sawaya, N.W., 2013. A discretization-based approach for the optimization of the multiperiod blend scheduling problem. Comp. Chem. Eng. 53, 122-142.

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A General Algorithm for Short-Term Scheduling of Batch Operations – 1. MILP formulation. Comp. Chem. Eng. 17, 211-227.

Kopanos, G. M.; Laínez, M. J.; and Puigjaner, L. 2009. An Efficient Mixed-Integer Linear Programming Scheduling Framework for Addressing Sequence-Dependent Setup Issues in Batch Plants. Industrial & Engineering Chemistry Research. 48, 6346–6357.

Kopanos, G. M., Puigjaner, L., Maravelias, C.T., 2011. Production Planning and Scheduling of Parallel Continuous Processes with Product Families. Ind. Eng. Chem. Res. 50, 1369–1378.

Kuhn, D.,Wiesemann,W., Georghiou, A., 2011. Primal and dual linear decision rules in stochastic and robust optimization. Mathematical Programming, 130(1), 177–209.

Lappas, N.H., Gounaris, C.E., 2016. Multi-stage adjustable robust optimization for process scheduling under uncertainty. AIChE J. 62, 1646-1667.

Lee, H., Pinto, J.M., Grossman, I.E., Park, S., 1996. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. Ind. Eng. Chem. Res. 35, 1630-1641.

Li, J., Karimi, I.A., 2011. Scheduling Gaoline Blending Operations from Recipe Determination to Shipping Using Unit Slots. Ind. Eng. Chem. Res. 50, 9156-9174.

Li, J., Li, W., Karimi, I.A., Srinivasan, R., 2007. Improving the Robustness and Efficiency of Crude Scheduling Algorithms. AIChE J. 53, 2659-2680.

Li, J., Xiao, X., Floudas, C.A., 2016. Integrated Gasoline Blending and Order Delivery Operations: Part I. Short-Term Scheduling and Global Optimization for Single and Multi-Period Operations. AIChE j. 62, 2043-2070.

Li, Z., Ierapetritou, M.G., 2009. Integrated production planning and scheduling using a decomposition framework. Chem. Eng. Sci. 64, 3585-3597.

Lima, R.M., Grossmann, I.E., Jiao, Y., 2011. Long-Term Scheduling of a Single-Unit Multi-Product Continuous Process to Manufacture High Performance Glass. Comp. Chem. Eng. 35, 554–574.

Lin, X., Floudas, C.A., Modi, S., Juhasz, N.M., 2002. Continuous-time Optimization Approach for Medium-range production scheduling of a multiproduct batch plant. Ind. Eng. Chem. Res. 41, 3884-3906.

Linnhoff, B., Townsend, D.W., Boland, D., Hewitt, G.F., Thomas, B.E.A., Guy, A.R., Marsland, R.H., 1982. A User Guide on Process Integration for the Efficient Use of Energy. The Institution of Chemical Engineers: Rugby, England.

Liu, Y., Karimi, I.A., 2007. Scheduling multistage, multiproduct batch plants with nonidentical parallel units and unlimited intermediate storage. Chem. Eng. Sci. 62, 1549-1566.

Liu, Y., Karimi, I.A., 2008. Scheduling multistage batch plants with parallel units and no interstage storage. Comp. Chem. Eng. 32, 671-693.

Liu, S., Pinto, J. M., Papageorgiou, L.G., 2008. A TSP-based MILP Model for Medium-Term Planning of Single-Stage Continuous Multiproduct Plants. Ind. Eng. Chem. Res. 47, 7733–7743.

Lotero, I., Trespalacios, F., Grossmann, I.E., Papageorgiou, D.J., Cheon, M.S., 2016. An MILP-MINLP decomposition method for the global optimization of a source based model for the multiperiod blending problem. Comp. Chem. Eng. 87, 13-35.

Magatão, L., Arruda, L.V.R., Neves, F.A., 2004. A mixed integer programming approach for scheduling commodities in a pipeline. Comput. Chem. Eng., 28, 171-185.

Maravelias, C.T., 2005. Mixed-Time Representation for State-Task Network Models. Ind. Eng. Chem. Res. 44, 9129-9145.

Maravelias, C.T., Grossmann, I.E., 2003. A new general continuous-time state task network formulation for short term, scheduling of multipurpose batch plants. Ind. Eng. Chem. Res. 42, 3056–3074.

Maravelias, C.T., Sung, C., 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. Comp. Chem. Eng. 33, 1919-1930.

Maudeli, A.M., Rippin, D.W.T., 1979. Production Planning and Scheduling for Multipurpose Batch Chemical Plants. Comp. Chem. Eng. 3, 199-206.

McCormick, G.P., 1976. Computability of global solutions to factorable nonconvex programs. Part I. Convez underestimating problems. Math. Program. 10, 147-175.

Méndez, C.A., Cerdá, J., 2003 An MILP continuous-time framework for short- term scheduling of multipurpose batch processes under different operation strategies. Optimization and Engineering, 4, 7–22.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. 30, 913-946.

Méndez, C.A., Henning, G. P., Cerdá, J., 2000. Optimal scheduling of batch plants satisfying multiple product orders with different due-dates. Comp. Chem. Eng. 24, 2223–2245.

Méndez, C.A., Henning, G.P., Cerdá, J., 2001. An MILP continuous time approach to short-term scheduling of resource constrained multi-stage flowshop batch facilities. Comput. Chem. Eng. 25, 701–711.

Meyer, C.A., Floudas, C.A., 2006. Global optimization of a combinatorially complex generalized pooling problem. AIChE J. 52, 1027-1037.

MirHassani, S.A., Jahromi, H.F., 2011. Scheduling multi-product tree-structure pipelines. Comput. Chem. Eng. 35, 165–176.

Misener, R., Floudas, C.A., 2014. ANTIGONE: Algorithms for continuous/integer Global Optimization of Nonlinear Equations. J. Glob. Optim. 59, 503-526.

Mostafaei, H., Castro, P.M., 2017. Continuous-time scheduling formulation for straight pipelines. AIChE J. 63, 1923-1936.

Mostafaei, H., Castro, P.M., Ghaffari-Hadigheh, A., 2015. A novel monolithic MILP framework for lot- sizing and scheduling of multiproduct treelike pipeline networks. Ind. Eng. Chem. Res. 54, 9202-9221.

Pantelides, C.C., 1994. Unified Frameworks for the Optimal Process Planning and Scheduling. In Proceedings of the Second Conference on Foundations of Computer Aided Operations; Cache Publications: New York, pp 253.

Papageorgiou, L.G., Shah, N., Pantelides, C.C., 1994. Optimal scheduling of heat-integrated multipurpose plants. Ind. Eng. Chem. Res. 33, 3168–86.

Raman, R., Grossmann, I.E., 1991. Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis. Comput. Chem. Eng. 15, 73-84.

Raman, R., Grossmann, I.E., 1994. Modeling and computational techniques for logic based integer programming. Comput. Chem. Eng. 18, 563-578.

Rejowski, R., Pinto, J.M., 2003. Scheduling of a multiproduct pipeline system. Comput. Chem. Eng. 27, 1229–1246.

Reklaitis, G.V., 1978. Review of scheduling of process operations. AIChE Symp. Ser. 78, 119-133.

Schilling, G., Pantelides, C.C., 1996. A Simple Continuous-Time Process Scheduling Formulation and a Novel Solution Algorithm. Comput. Chem. Eng. 20, S1221-S1226.

Relvas, S., Matos, H.M., Barbosa-Póvoa, A.P.F.D., Fialho, J., Pinheiro, A.S., 2006. Pipeline scheduling and inventory management of a multiproduct distribution oil system. Ind. Eng. Chem. Res. 45, 7841–7855.

Seid, E.R., Majozi, T., 2014. Heat integration in multipurpose batch plants using a robust scheduling framework. Energy, 71, 302-320.

Shah, N., 2004. Pharmaceutical supply chains: key issues and strategies for optimization. Comp. Chem. Eng. 28, 929–941.

Shah, N., Z. Li, and M.G. Ierapetritou, Petroleum Refining Operations: Key Issues, Advances, and Opportunities, Industrial and Engineering Chemistry Research 50 (3), 1161(2011).

Shah, N., Pantelides, C.C., Sargent, R.W.H., 1993. A General Algorithm for Short-Term Scheduling of Batch Operations – 2. Computational Issues. Comp. Chem. Eng. 17, 229-244.

Shaik, M., Floudas, C.A., 2008. Unit-specific event-based continuous-time approach for short-term scheduling of batch plants using RTN framework. Comput. Chem. Eng. 32, 260–274.

Shah, N., 1998. Single- and multisite planning and scheduling: current status and future challenges. Proceedings of FOCAPO-98, pp 75-90.

Singh, A., Forbes, J.F., Vermeer, P.J., Woo, S.S., 2000. Model-based real-time optimization of automotive gasoline blending operations. J. Process Control 10, 43-58.

Sundaramoorthy, A., Karimi, I.A., 2005. A simpler better slot-based continuous- time formulation for short-term scheduling in multipurpose batch plants. Chem. Eng. Sci. 60, 2679–2702.

Sundaramoorthy, A., Maravelias, C.T., 2011. Computational Study of Network-based Mixed-Integer Programming Approaches for Chemical Production Scheduling. Ind. Eng. Chem. Res. 50, 5023-5040.

Sung, C., Maravelias, C.T., 2007. Production Planning in Process Systems Engineering. In Pistikopoulos, E. N.; Georgiadis, M. C.; Dua, V.; and Papageorgiou, L. G., eds., Process Systems Engineering: Supply Chain Optimization, volume 4. Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA. chapter 9.

Sung, C., Maravelias, C.T., 2009. A Projection-Based Method for Production Planning of Multiproduct Facilities. AIChE J. 55, 2614-2630.

Susarla, N., Li, J., Karimi, I.A., 2010. A novel approach to scheduling multipurpose batch plants using unit-slots. AIChE J. 56, 1859–1879.

Swaney, R.E., Grossmann, I.E., 1985. An Index for Operational Flexibility in Chemical Process Design – Part I: Formulation and Theory. AIChE J. 31(4), 621–630.

Tawarmalani, M., Sahinidis, N.V., 2002. Convexification and Global Optimization in Continuous and Mixed Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Dordrecht: Kluwer Academic Publishers. pp 254-284.

Tawarmalani, M., Sahinidis, N.V. A polyhedral branch-and-cut approach to global optimization. Math. Program. 103, 225-249.

Teles, J.P., Castro, P.M., Matos, H.A., 2013. Multiparametric disaggregation technique for the global optimization of polynomial programming problems. J. Glob. Optim. 55, 227-251.

Terrazas-Moreno, S., Grossmann, I.E., 2011a. A multiscale decomposition method for the optimal planning and scheduling of multisite continuous multiproduct plants. Chem. Eng. Sci. 66, 4307–4318.

Terrazas-Moreno, S., Trotter, P., Grossmann, I.E., 2011b. Temporal and spatial Lagrangean decompositions in multi-site, multi-period production planning problems with sequence-dependent changeovers. Comp. Chem. Eng. 35, 2913– 2928.

Trespalacios, F., Grossmann, I.E., 2014. Review of Mixed-Integer Nonlinear and Generalized Disjunctive Programming Methods. Chem. Ing. Tech. 86, 991–1012.

Trespalacios, F. and I.E. Grossmann. 2015. Improved Big-M Reformulation for Generalized Disjunctive programs. Computers & Chemical Engineering, 76, 98–103.

Velez S., Maravelias C.T., 2015. Theoretical Framework for the Formulation of MIP Scheduling Models with Multiple and Nonuniform Discrete-time Grids. Comput. Chem. Eng. 72, 233-254.

Velez S., Maravelias C.T., 2013. Multiple and Nonuniform Time Grids in Discrete-Time MIP Models for Chemical Production Scheduling. Comput. Chem. Eng. 53, 70-85.

Verderame, P.M., Elia, J.A., Li, J., Floudas, C.A., 2010. Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors. Ind. Eng. Chem. Res. 49, 3993–4017.

Wassick, J., 2009. Enterprise-wide optimization in an integrated chemical complex. Comput. Chem. Eng. 33, 1950-1963.

Westenberger, H., Kallrath, J., 1994. Formulation of a job shop problem in process industry. Working paper, Bayer AG, Leverkusen and BASF AG, Ludwigshafen, Germany

Wicaksono, D.N., Karimi, I.A. Piecewise MILP under- and overestimators for global optimization of bilinear programs. AIChE J. 54, 991-1008.

Yee, T.F., Grossmann, I.E., Kravanja, Z., 1990. Simultaneous Optimization Models for Heat Integration-1. Area and Energy Targeting and Modeling of Multistream Heat Exchangers. Comput. Chem. Eng. 14, 1151-1164.

Zhang, Q., Grossmann, I.E., 2016. Enterprise-wide optimization for industrial demand side management: Fundamentals, advances, and perspectives. Chemical Engineering Research & Design, 116, 114-131.

Zhang, Q., Grossmann, I.E., Heuberger, C.F., Sundaramoorthy, A., Pinto, J.M., 2015. Air Separation with Cryogenic Energy Storage: Optimal Scheduling Considering Electric Energy and Reserve Markets. AIChE Journal, 61(5), 1547-1558.

Zhang, Q., Sundaramoorthy, A., Grossmann, I.E., Pinto, J.M., 2016. A discrete-time scheduling model for continuous power-intensive process networks with various power contracts. Computers & Chemical Engineering, 84, 382-393.

Zhang, Q., Morari, M.F., Grossmann, I.E., Sundaramoorthy, A., Pinto, J.M., 2016b. An adjustable robust optimization approach to scheduling of continuous industrial processes providing interruptible load. Computers & Chemical Engineering, 86, 106-119.

Zhang, Q., Lima, R.M., Grossmann, I.E., 2016c. On the Relation Between Flexibility Analysis and Robust Optimization for Linear Systems. AIChE Journal, 62(9), 3109-3123.

Zyngier, D., Kelly, J.D., 2012. UOPSS: A New Paradigm for Modeling Production Planning & Scheduling Systems. In Proceedings of the 22nd European Symposium on Computer Aided Process Engineering. Edited by Ian Bogle and Michael Fairweather.