OR Spectrum (2002) 24: 219–250

OR Spectrum

© Springer-Verlag 2002

Planning and scheduling in the process industry

Josef Kallrath[1,2]

[1] BASF-AG, GVC/S (Scientific Computing) - C13, 67056 Ludwigshafen, Germany
(e-mail: josef.kallrath@basf-ag.de)
[2] Astronomy Department, University of Florida, Gainesville, FL 32661, USA
(e-mail: kallrath@astro.ufl.edu)

Abstract. Since there has been tremendous progress in planning and scheduling in the process industry during the last 20 years, it might be worthwhile to give an overview of the current state-of-the-art of planning and scheduling problems in the chemical process industry. This is the purpose of the current review which has the following structure: we start with some conceptional thoughts and some comments on special features of planning and scheduling problems in the process industry. In Section 2 the focus is on planning problems while in Section 3 different types of scheduling problems are discussed. Section 4 presents some solution approaches especially those applied to a benchmark problem which has received considerable interest during the last years. Section 5 allows a short view into the future of planning and scheduling. In the appendix we describe the Westenberger-Kallrath problem which has already been used extensively as a benchmark problem for planning and scheduling in the process industry.

Key words: Mixed integer programming – Supply chain optimization – Process industry – Planning – Scheduling

1 Introduction

1.1 Special features in the process industry

In the process industry *continuous* and *batch production systems* can be distinguished. There exists also semi-batch production which combines features from both. Plants producing only a limited number of products each in relatively high volume typically use special purpose equipment allowing a continuous flow of materials in *long campaigns*, i.e., there is a continuous stream of input and output products with no clearly defined start or end time. Alternatively, small quantities of a large number of products are preferably produced using multi-purpose equipment which are operated in *batch mode,* i.e., there is a *well-defined start-up,* e.g., filling in some products, well-defined follow-up steps defined by specific recipes, e.g., heating the product, adding other products and let them react, and a *clearly defined end*, e.g., extracting the finished product. Batch production involves an integer number of batches where a batch is the smallest quantity to be produced, e.g., 500 kg. Several batches of the same product following each other immediately establish a *campaign*. Production may be subject to certain constraints, e.g., campaigns are built up by a discrete number of batches, or a minimal campaign length (or production quantity) has to be observed. Within a fixed planning horizon, a certain product can be produced in several campaigns; this implies that campaigns have to be modelled as individual entities.

Another special feature in the refinery or petrochemical industry or process industry in general is the *pooling* problem (see, for instance, [28], or Chapter 11 in [42]), an almost classical problem in nonlinear optimization. It is also known as the *fuel mixture problem* in the refinery industry but it also occurs in blending problems in the food industry. The pooling problem refers to the intrinsic nonlinear problem of forcing the same (unknown) fractional composition of multi-component streams emerging from a pool, e.g., a tank or a splitter in a mass flow network. Structurally, this problem contains indefinite bilinear terms (products of variables) appearing in equality constraints, e.g., mass balances. The pooling problem occurs in all multi-component network flow problems in which the conservation of both mass flow and composition is required and both the flow and composition quantities are variable.

Non-linear programming (NLP) models have been used by the refining, chemical and other process industries for many years. These nonlinear problems are non-convex and either approximated by linear ones and solved by linear programming (LP) or approximated by a sequence of linear models. This sequential linear programming (SLP) technique is well established in the refinery industry but suffers from the drawback of yielding only locally optimum solutions. Although many users may identify obviously sub-optimal solutions from experience, there is no validation of those which are not obviously so, as this would require truly globally optimal solutions. From an end-user point of view, the problems of existing technology are becoming ever more acute. Since the market for products such as gasoline and chemicals are becoming increasingly amalgamated, many planning problems now necessarily involve multiple production facilities in geographically separate sites, with concomitant interactions and interconnections. These are hard to solve and much more prone to giving sub-optimal local solutions, particularly if they stretch over many time periods. However, recent advances in optimization algorithms have yielded experimental academic codes which do find truly globally optimal solutions to these NLP models. Non-convex nonlinear models are not restricted to the oil refining and petrochemical sector, but arise in logistics, network design, energy, environment, and waste management as well as finance and their solution asks for global optimization.

In the chemical process industry, the proper description of the reaction kinetics leads to exponential terms. If, in addition, plants operate in discrete modes or connections between various units, e.g., tanks and crackers or vacuum columns have to be chosen selectively, then mixed-integer nonlinear optimization problems need to be solved. Process network flow or process synthesis problems [30] usually fall into this category, too. Examples are heat exchanger or mass exchange networks.

Planning and scheduling is part of company-wide logistics and supply chain management. However, to distinguish between those topics, or even to distinguish between planning and scheduling is often a rather artificial approach. In reality, the border lines between all those areas are diffuse. There are strong overlaps between scheduling and planning in production, distribution or supply chain management and strategic planning.

1.2 Some comments on planning and scheduling in the process industry

Although the boundary between planning and scheduling is diffuse let us try to work out a few structural elements of planning and scheduling, which may include the following features:

- multi-purpose (multi-product, multi-mode) reactors,
- sequence-dependent set-up times and cleaning cost,
- combined divergent, convergent and cyclic material flows,
- non-preemptive processes (no-interruption), buffer times,
- multi-stage, batch & campaign production using shared intermediates,
- multi-component flow and nonlinear blending,
- finite intermediate storage, dedicated and variable tanks.

Structurally, these features often lead to allocation and sequencing problems, knapsack structures, or to the pooling problem. As there is no clear definition of the border line between planning and scheduling problems, we try to illuminate the subject from different angles by summarizing a few aspects and objectives of planning and scheduling and try to develop a kind of an informal definition serving as a platform. In production or supply chain planning, we usually consider material flow and balance equations connecting sources and sinks of a supply network. Time-indexed models using a relative coarse discretization of time, *e.g.*, a year, quarters, months or weeks are usually accurate enough. LP, MILP and MINLP technologies are often appropriate and successful for problems with a clear quantitative objective function as outlined in Section 2, or quantitative multi-criteria objectives.

In scheduling problems the focus on time is more detailed and may require even continuous time formulations. Furthermore, one faces rather (conflicting) goals than objectives: the optimal use of resources, minimal makespan, minimal operating cost or maximum profit versus more qualitative goals such as reliability (meet demand in time, proper quality, etc.) and robustness; such qualitative goals are often hard to quantify. The short-term operational aspects of operating a set of chemical reactors, food producing machines or distillation columns in a refinery are of primary interest. Users are mostly interested in feasible, acceptable and robust schedules, the objectives are usually somewhat vague, but it is common that the possibility to interact and to re-schedule, as well as the stability of solutions in cases of re-scheduling are highly appreciated. Scheduling problems are usually NP-hard, no standard solution techniques are available and, actually, in many cases we are facing feasibility problems rather than optimization problems. The solution approaches found in the literature are:

- *exact* and *deterministic methods* such as mathematical optimization including MILP and MINLP, graph theory (GT) or constraint programming (CP), or *hybrid approaches* in which MILP and CP are integrated,
- *meta-heuristics* (evolutionary strategies, tabu search, simulated annealing, ...) as described briefly in Section 4.2.4 or [48].

In addition to these remarks it is worthwhile to comment on the difference between *offline* and *online scheduling* [64]. Offline scheduling as mostly discussed in this article, in the ideal case, assumes that all data of a problem are given, i.e., full knowledge of the future (of course, this is also an approximation since our knowledge of future demand or orders is uncertain), and is close to planning except for the length of the time horizon and the resolution of time. Online scheduling as a special case of (combinatorial) on-line optimization [11] makes decisions based on past events and current data without information about future events relevant for the current decision problem; many decisions have to be made before all data are available and decisions once made cannot be changed. It may involve current process control data, updated demand data and orders, but misses orders which may enter the system in the near future and within the horizon of the current schedule to be determined. The goal is to exploit uncertain (w.r.t. the future) and incomplete information in such a way to improve the final quality of its overall performance, i.e., the quality of schedules over rolling time horizons. Unlike in stochastic optimization, where known data are subject to stochastistic uncertainties, the uncertainty in on-line scheduling only arises from the uncertainty of future data.

2 Model features in planning problems

Planning in the process industry is used to create production, distribution, sales and inventory plans based on customer and market information while observing all relevant constraints. In particular, operational plans have to be determined which are aimed to structure future production, distribution and other related activities according to business objectives. It is common practice that, based on these operational plans, detailed schedules are worked out which define the precise timing and sequencing of individual operations as well as the assignment of the required resources over time. Planning tools and software packages from various vendors are designed to incorporate new market and operational information quickly and help business users to keep their operations performing at their optimum. Especially, nowadays it is possible to find the optimal way to meet business objectives and to fulfill all production, logistics, marketing, financial and customer constraints and especially

- to accurately model *single site* and *multi-site networks*;
- to perform *capital planning* and *acquisition or divestiture analysis*, i.e., to have the possibility to change the structure of a manufacturing production network through investment and to determine the best investment type, size and location

based on user defined rules relating to business objectives and available resources, e.g., Kallrath [40]; the results of such analysis can lead to non-intuitive solutions that provide management with scenarios that could dramatically increase profits;

- to produce *integrated enterprise solutions* and to enable a cross-functional view of the planning process involving production, distribution and transport, sales, marketing and finance functions.

Planning as part of the supply chain management may focus on short and mid-term *sales and operations planning* or long-term *acquisition, consolidation, and capacity analysis* with a strategic focus. In the literature and in available software packages we usually find *time-indexed models* supporting multi-period analysis, i.e., nearly all the data may vary over time and allow to evaluate scenarios that involve time dependent aspects such as seasonal demand patterns, new product introductions, shutdown of production facilities for maintenance periods. These models include the following main structural objects:

- **Locations** can be production or storage sites, hosting plants and tanks, or demand points hosting tanks.
- **Facilities** typical are production, wrapping or inventory units that are characterized by their functional properties. Especially, in the process industry we find multi-stage production systems involving units with general product-mode relationships. Their functional properties are attributes such as capacity, throughput rates, product recipes, yields, minimum production utilization rates, fixed and variable costs, or storage limitations. Facilities can be existing or potential (for design studies). Production facilities may be subject to batch and campaign constraints across periods.
- **Demand Points** may represent customers, regional warehouse locations or distributors who specify the quantity of a product they request. A demand point can be also seen as a sink of the planning model, i.e., a point where a product leaves the system and is not further traced. Demand may be subject to certain constraints, e.g., satisfying a minimum quantity of demand, observing origins of production or supplying a customer always from the same origin.
- **Inventories** may be physically fixed entities such as tanks or warehouses but also moveable entities (e.g., drums, containers, boxes, etc.). They can be defined as dedicated

  1. to a single product from one production source,
  2. dedicated to a specific product, or
  3. free to accept any product from any source or origin.

  We may encounter tank farms, and especially *multi-purpose storage entities*, i.e., variable and multi-product tanks.
- **Products** may be classified as raw materials, intermediates, finished and salable products. A product may have several of these attributes, and it can be purchased from suppliers, produced or sold. Products are produced according to the capabilities at the facilities and the recipes assigned. Products may establish a product group, e.g., additives. Product requirements are based on market demand which is characterized by volume, selling price, package type, time,

origin and location or by other products in which they are used as intermediate products.
- **Suppliers** or **vendors** may provide products for purchase under different offering schemes. This includes the ability to link the product supply to locations and describe contractual pricing mechanisms or availability. The solver may choose the optimal supplier.

Regarding the overall business and strategic objectives the model needs to incorporate data describing the

- costs, i.e., certain fixed costs, variable costs (production, transportation, inventory, external product purchase, energy, resources and utilities), and further
- commercial aspects: financial aspects such as depreciation plans, discount rates, investment plans, foreign currency exchange rates, duties and tariffs, as well as site dependent taxes.

Maximize operating cash flow and *maximize net present value* (NPV) objective functions are used to determine the financial and operating impacts of mergers, acquisitions, consolidation initiatives, and capital spending programs effecting business. In detail this may include:

1. maximize the net profit (free design reactors; open and close facilities),
2. maximize the contribution margin for a fixed system of production units,
3. maximize the contribution margin while satisfying a minimum percentage of demand,
4. minimize the cost while satisfying full demand (allow external purchase of products),
5. maximize total sales neglecting cost,
6. maximize total production for a fixed system of production reactors,
7. maximize total production of products for which demand exists,
8. minimize energy consumption or the usage of other utilities,
9. minimize the deviation of the usage of resources from their average usage,
10. multi-criteria objectives, e.g., maximize contribution margin and minimize total transportion volume.

Planning involves the determination of operational plans that support different short- or mid-term objectives for the current business. By using different objective functions it is possible to create operational plans that support strategies such as market penetration, top-line growth, or maximization of cash flow to support other business initiatives.

If, besides this broad structure, the focus is on a more detailed representation of physical entities, we find that planning models and their constraints may involve the following features (in alphabetic order):

- **Batch production**: The quantity of a specific product being produced in a campaign possibly over several periods must be an integer multiple of some pre-defined batch size.
- **Buy, build, close or sell specific production assets**: This feature is used for closing, or selling acquisition, consolidation and capacity planning to determine

the NPV and operational impacts of adding or removing specific assets or groups of assets to the network.

- **Campaign production**: This allows to impose a lower and/or an upper bound on a contiguous production run (campaign) possibly across periods; this feature is also known under the name *minimal runs*.
- **Delay cost**: Penalty cost apply if customer orders are delivered after the requested delivery date.
- **Minimum production requirements**: Minimum utilization rates modelled as semi-continuous variables have to be observed for specific production units and/or entire production locations for each production time period.
- **Multi-locations**: This can be production sites, storage sites, and demand points.
- **Multi-purpose production units**: If a unit is fixed to a certain mode, several products are produced (with different mode-dependent daily production rates), and vice-versa, a product can be produced in different modes. Daily production can be less than the capacity rates. A detailed mode-changing production scheme may be used to describe the cost and time required for sequence-dependent mode-changes.
- **Multi-stage production**: Free and fixed recipe structures allow for the production of multiple intermediate products before the manufacture of the final product with convergent and divergent product flows. The recipes may depend on the mode of the multi-purpose production unit.
- **Multi-time periods**: Non-equidistant time period scales are possible for commercial and production needs. For instance, demand may be forecast weekly for the first quarter of the year and then quarterly for the remainder of the year.
- **Nonlinear pricing** for the purchase of products or utilities (energy, water, etc.) or nonlinear cost for inventory or transportation may lead to convex and concave structures in order to model volume and price discount schemes for the products or services purchased, while in addition, contract start-up and cancellation fees may lead to additional binary variables.
- **Order lost cost**: Penalty cost are incurred if products are not delivered as requested and promised.
- **Packaging** machines are optimized to increase machine throughput and assure that priority is given to the most profitable products.
- **Product swaps**: With the objective of saving transportation and other cost companies often arrange joint supply agreements called *swaps*. For example: Company 1 based in Europe as well in the US has a production shortage of product A in the US and thus purchases a defined quantity of product A in the US from company 2. Company 2 (also located in the US and Europe) has a customer in Europe requesting product A and thus purchases a defined quantity product A from company 1 in Europe. Both companies get product A where they need it and avoid the cost of shipping the product. Without this type of supply agreement company 1 would have to ship product A from its European plant to the US, and company 2 would have to ship product A from its US manufacturing plant to Europe.
- **Production origin tracing**: It is possible to define fixed, free or unique origins for specific demands. For example, a customer may require that his demand is

satisfied only from a specific plant in the network, or it may not be supplied from a set of plants, or the customer only requests that he is supplied from one unique plant during the whole planning horizon.
- **Shelf-life time**: Product aging time can be traced. This allows for the application of constraints such as: maximum shelf-life time, disposal costs for time expired products, and the setting of selling prices as a function of product life.
- **Transportation** and **logistics**: Transportation quantities are appropriately modelled by the use of semi-continuous variables. This allows minimum and maximum shipment quantities to be defined for each source location, destination location, product, and transport mean combination. The logistics involves the costs and lead times and constraints (minimum shipment quantities) associated with moving intermediate and finished products between facilities and demand points. The mean of transport may be chosen by the optimizer and nonlinear cost functions have to be considered as well.

This list covers many features but may be incomplete.

3 Types of scheduling problems in the process industry

In this review, we present some typical structures and categories of scheduling problems. Their treatment differs by the mathematical techniques applied:

- batch and campaign planning,
- scheduling problems in the chemical process industry including lot-sizing and sequencing,
- time-precedence and aggregate resource constraints,
- nonlinear scheduling problems including blending.

3.1 Batch and campaign planning

For a given planning problem formulated as a time-indexed model, it may be desirable to include certain constraints in the model which allow to incorporate batch and campaign features. To be as general as possible we might want to consider batch reactors which can be, for example, operated in different modes producing, at a given time, several products in each mode with different free or fixed recipes leading to a general mode-product relation ([42], pp.153–155, 320–324). Thus, in a certain mode several products are produced (with different daily production rates), and vice-versa, a product can be produced in different modes. Daily production can be less than the capacity rates. This is an important feature in demand-driven joint production in which several products are produced in fixed or variable ratios to each other.

In time-indexed model formulations where variables p_{pt} define the production quantity [e.g., in tons] of a product p in period (time-interval) t it is not easy to model batch or campaign restrictions if the batch or minimal campaign size is larger than the capacity per period. Assume that production is performed in batches of 200 tons, and that the time intervals have a length of ten days with a daily production rate of 10 tons/day. The minimum time to produce the batch would cover 20 days, or exactly two time intervals. A plan looking like $p_{p4} = 45$ tons, $p_{p5} = 100$ tons, and $p_{p6} = 55$ tons covers three periods (the first and third only partial) to produce exactly 200 tons, and thus provides more degrees of freedom.

Brockmüller and Wolsey [12] solved the problem for a special case (daily production equals the capacity rates). Their approach, which falls into the class of discrete lotsizing and scheduling problems (DLSP, [20, p.146]), uses explicitly the feature that production equals the capacity rates in order to determine a priori the number of periods to produce a campaign of specified minimal size. However, if daily production can take any value between a lower bound, e.g., zero, and the capacity rate per day, or if a product is produced, for example, according to general mode-product relations, then this a priori information is not available. Kallrath [38] overcomes this restriction and has developed an extension which can be added to any production planning model formulated as a proportional lotsizing and scheduling problem (PLSP, [24]), i.e., based on time-indexed formulations with at most one setup- or mode-change per period. It allows to add constraints involving accumulated quantities over several time-slices thus implementing the concept of contiguity into the model. This feature is relevant to any kind of process industry. It allows to model batch and campaign production or to require that a certain time-lag between successive mode-changes is observed. The key idea used in the technical approach is to identify which time-indexed production contributions belong to certain contiguous components, e.g., campaigns, over several time slices and to replace products of continuous and binary variables, or absolute value terms by linear relations involving additional binary variables. This approach has successfully been applied to production planning problems in the chemical industry. Other approaches are described by [31], [7] or [47].

3.2 A typical scheduling problem in the chemical industry

Westenberger and Kallrath (1994) in a cooperational work of Bayer AG and BASF Aktiengesellschaft formulated a typical but generic scheduling problem with the objective to push the development of algorithms for scheduling problems in process industry. Their proposal to establish a working group to develop standardized benchmark problems for plannig and scheduling in the chemical industry initiated many research projects and activities. The *Westenberger-Kallrath problem* has been understood as a typical scheduling problem occurring in process industry including the major characteristics of a real batch production process (involving multi-product facilities, multi-stage production, combined divergent and convergent product flows, variable batch sizes, non-preemptive processes, shared intermediates, alternative recipes, flexible proportions of output products, blending processes, sequence and usage dependent cleaning operations, finite intermediate storage, cyclic material flows, re-usage of carrier substances, and no-wait production for certain types of products) so as to encourage researchers and engineers to test their algorithms and software tools by applying them to this test case. Solutions based on mathematical programming techniques, heuristics, simulation, genetic algorithms, evolutionary strategies etc. may be sent to the author. Contributions and results are collected under http://www.math.tu-berlin.de/chemical-benchmarks/; the problem is summarized in the appendix of this review paper. A mathematical description has been given in order to guarantee a unique interpretation of the test problem. The problem description contains all data necessary to perform own test calculations; no special chemical knowledge is needed. Everybody should feel free to transform it to his needs as long as equivalency is guaranteed. Mathematical methods and typical results for related problems can be found in literature [21], [41], [60], [17].

The test and benchmark problem includes all features described above and should be solved for a set of different objective functions among them:

1. minimization of makespan,
2. maximization of profit by optimizing the product mix,
3. minimization of investment cost by optimization of stock and production capacities.

Over the last 6 years many publications ([3], [8], [10], [14], [15], [43], [54], [58], [63], [69], [9], [52]) appeared in which different techniques have been applied to produce solutions to the problem; some results are summarized in [52].

3.3 Time-precedence and aggregate resource constraints

For a general discussion of such problems see, for instance, [56], [53] and [13]. A complex scheduling problem in the process industry [42] containing precedence and aggregate resource constraints has been tackled by many authors and is briefly summarized below.

The client uses a set of machines and employs a number of workers. He receives orders from his customers. Each order demands a certain quantity of a product which can be produced on the client's machines. The machines are operated and supervised by the workers. Orders are often split up into several identical jobs, which are necessary in order to produce the required quantity of the product, because, typically, orders demand a larger quantity of a specific product than the machine capacity allows to produce in one job. A job for a given order is processed on a machine according to a specific *procedure* or *process plan*. It consists of a deterministic sequence of tasks defining how to produce a specific product. The size of a job is limited by the capacity of the machine. Each task has a pre-defined demand for labour and a certain duration. The workers are allocated to different tasks in order to keep the jobs running. Allocation of the workers has to comply with working regulation rules, e.g., taking breaks, washing, equally spread labour among the workers, limits on labour intensive work, over-occupation rate and over-time. The objective is to minimize the makespan and/or to minimize the (variation in the) number of workers. In [42] we find a model including assignment and sequencing decisions, and a time-indexed formulation to describe the detailed personnel requirements. Another solution technique to solve the problem is described in Section 4.2.3.

3.4 Nonlinear scheduling problems in refineries

Planning and scheduling has a long history in the refinery industry and dates back to the 1950s. If treated in the context of mathematical optimization, refinery scheduling leads to MINLP problems due to the presence of the pooling problem. It is common practice in the refinery industry that the scheduling problem usually comes up after a production planning problem based on mass balances has been solved for a medium time horizon, consisting of monthly or quarterly periods. The data generated by the production planning problem are input data to the scheduling problem which determines a detailed crude oil processing schedule, process unit schedule and blending and shipping schedules. The purpose of the scheduling problem is to transform the production plan into a schedule useful for all operations within a time horizon of a few days. In that sense the scheduling problem is rather a feasibility problem than an optimization problem. While in typical production scheduling problems degeneracy and symmetry cause large problems, the nonlinear features in the refinery scheduling problems could destroy some symmetry and may lead to useful relaxations.

A typical example is a medium-sized refinery which schedules the production and the storage of oil products for the next two to four weeks. The input data for the scheduling problem are provided by production plans generated by solving an LP problem. The model describes the typical processes in a refinery, e.g., the flows of crude oil from tankers to crude oil tanks, from the tanks to the initial production units, i.e., top distillation units, and the distribution to intermediate tanks or to further production units, e.g., vacuum distiller or thermal cracker, and finally, filling the product tanks and the delivery of the final products. The model takes into account that there are several types of oils and blending components (intermediate products) with different chemical or physical properties and that they are treated differently by the production units. In addition the model considers minimum and maximum capacity restrictions for tanks and production units. Besides this the units require a minimal run time period once they are activated.

The model includes logical constraints determining the daily production schedule and the assignment of raw material and intermediates to tanks and production units. The objective of the optimization problem is to minimize deviations from targets, e.g., deviations from the pre-determined production plan, or, alternatively, to minimize the use of blending components instead of crude oil products. For a case study within a confidential research project, the model included 16 crude oils, 4 blending components, 17 crude oil tanks, 5 production units and 8 final products.

The features and the constraints of the refinery model described above can be summarized and classified into the following groups:

- flow of oils and blending components (linear mass balance equations),
- quality constraints and capacity limits of production units and tanks (inequalities),
- proportional composition of streams (nonlinear equations),
- assignment of tanks and production units (equation & inequalities involving binary variables).

For a 30-days scenario the model contains 80,000 constraints, 70,000 continuous variables, 2,500 binary variables and 130,000 nonzero entries in the matrix. The problem is very difficult to solve not only because of its size but, primarily, because of the combination of *nonlinear constraints* and *binary variables*. The nonlinear character of the mathematical model is caused by the fact that different types of crude oils or blending components of different properties are mixed, whereas the decisions on the daily production schedule and the assignment of crude oils and intermediates to tank and production units require the use of binary variables.

The solution of this MINLP problem has been derived using the software package `XPRESS-MP` by Dash Optimization. For this problem, Dash had extended their Branch&Bound-facilities by a special recursion algorithm for the nonlinear terms. Solutions have been obtained on a PC with Pentium I processor within acceptable time (approximately one hour for a 17-days scenario).

4 Solution approaches

4.1 Solution approaches used in planning

Most of the planning problems in the process industry lead to MILP or MINLP models and contain the following building blocks: *tracing the states of plants, modeling production, balance equations for material flows, transportation terms, consumption of utilities, cost terms*, and *special model features*. Mode-changes, start-up and cancellation features, and nonlinear cost structures require many binary variables. Minimum utilization rates and transportation often require semi-continuous variables. Special features such as batch and campaign constraints across periods require special constraints to implement the concept of contiguity. The model, however, remains linear in all variables. Only if the pooling problem occurs, *e.g.*, in the refinery industry or the food industry, we are really facing a MINLP problem. For a review on algorithms used in LP, MILP, NLP, and MINLP the reader is referred to [39].

Using state-of-the art commercial solvers, e.g., `XPRESS-MP` [`XPRESS-MP` is by Dash Optimization, http://www.dashoptimization.com] or `CPLEX` [`CPLEX` is by ILOG, http://www.ilog.com], MILP problems can be solved quite efficiently. In the case of MINLP, the solution efficiency depends strongly on the individual problem and the model formulation. However, as stressed in [39] for both problem types, MILP and MINLP, it is recommended that the full mathematical structure of a problem is exploited, that appropriate reformulations of models are made and that problem specific valid inequalities or cuts are used. Software packages may also differ with respect to the ability of *pre-solving techniques*, *default-strategies* for the Branch&Bound algorithm, *cut generation* within the Branch&Cut algorithm, and last but not least *diagnosing and tracing infeasibilities* which is an important issue in practice.

Current activities to solve planning problems more efficiently are focused on the construction of useful valid inequalities for certain substructures of planning problems. Those inequalities may a priori be added to a model, and in the extreme case they would describe the complete convex hull. As an example we consider the mixed-integer inequality

$$x \leq C\lambda \quad , \quad 0 \leq x \leq X \quad ; \quad x \in \mathbb{R}_0^+ \quad , \quad \lambda \in \mathbb{N} \tag{1}$$

which has the valid inequality

$$x \leq X - G(K - \lambda) \quad \text{where } K := \left\lceil \frac{X}{C} \right\rceil \quad \text{and} \quad G := X - C\,(K - 1) \tag{2}$$

This valid inequality (2) is the more useful, the more K and X/C deviate. A special case arising often is the situation $\lambda \in \{0, 1\}$. Another example, taken from ([70], p. 129) is

$$A_1\alpha_1 + A_2\alpha_2 \leq B + x \qquad x \in \mathbb{R}_0^+ \qquad \alpha_1, \alpha_2 \in \mathbb{N} \tag{3}$$

which for $B \notin \mathbb{N}$ leads to the valid inequality

$$\lfloor A_1 \rfloor \alpha_1 + \left(\lfloor A_2 \rfloor \alpha_2 + \frac{f_2 - f}{1 - f} \right) \leq \lfloor B \rfloor + \frac{x}{1 - f} \tag{4}$$

where the following abbreviations are used:

$$f := B - \lfloor B \rfloor \quad , \quad f_1 := A_1 - \lfloor A_1 \rfloor \quad , \quad f_2 := A_2 - \lfloor A_2 \rfloor \tag{5}$$

The dynamic counterpart of valid inequalities added a priori to a model leads to cutting plane algorithms which avoid adding a large number of inequalities a priori to the model (note, this can be equivalent to finding the complete convex hull). Instead, only those useful in the vicinity of the optimal solution are added dynamically. For the topics of valid inequalities and cutting plane algorithms the reader is referred to the well written book by Wolsey [70].

Using these techniques, for some BASF planning problems including up to 100,000 constraints and up to 150,000 variables with several thousand binary variables, good solution with integrality gaps below 2% have been achieved within 30 minutes on standard Pentium machines [39].

4.2 Solution approaches used in scheduling

The complexity of scheduling problems can easily exceed today's hardware and algorithmic capabilities. Nevertheless, there are numerous promising contributions (see, for instance, [67], [45], [55], [4], [65], [63], [52], [69]) for problems in the process industry. What makes scheduling problems so difficult? Using exact methods such as MILP, in some cases it is not even possible to find feasible integer solutions because feasible integer solutions exist often only very deep in the B&B tree. In many cases it is very difficult to derive useful upper and/or lower bounds. Scheduling problems usually suffer from poor LP relaxations. Resource constraints can easily be fulfilled with fractional values of the binary variables used in time-indexed formulations and thus lower bounds are very weak. Even using parallel algorithms and powerful hardware, scheduling problems might be too complex, and often cannot be solved with MILP methods, at least not yet. If we meet such cases, it is also worthwhile to apply another exact method: constraint programming (CP). Heipcke [33] investigated successfully a very difficult scheduling problem (Section 10.5 in [42]), which became a benchmark problem in both the MIP and CP community as well as amongst scientist using graph theory or meta-heuristics. Heipcke originally applied both methods, CP versus MILP, to this problem and later contributed to the combination of both techniques [34]. Timpe [68] reports a successful application of a combined MIP-CP approach to a scheduling problem in the chemical industry. CP [46] has been developed in the 1980s out of Logic Programming and Constraint Solving and has been applied successfully to a large range of industrial applications, especially to discrete (optimization) problems. CP is a technique for discrete optimization that uses a tree search and performs domain reduction at each node. CP models typically include a wide range of constraint types, e.g., special global constraint operators such as *all different* or *cumulative*. Unfortunately, in the CP community the motivation is low to develop a common language which would allow to formulate a problem in a very compact way as is the case for mathematical programming languages. Instead, most and generic CP applications require the users to program in C or C++; this makes it very difficult to port the model to different hardware platforms and to maintain the software over a longer period.

If CP also fails, the last resort might be to use heuristic approaches [48], e.g., simulated annealing or tabu search. Heuristics exploiting the structure of scheduling chemical batch processes can also lead to good results as the two-stage solution procedure by Blömer and Günther [10]) demonstrates. In the first stage, an LP-based heuristic produces an initial solution. The proposed *time grid heuristic* defines a time grid that includes only a limited number of feasible periods in which a processing task is allowed to start. Thus, the size of the original multi-period MILP model is reduced in a controlled manner and optimal solutions of the relaxed model are obtained within reasonable computational time.

4.2.1 Processes and the state-task network (STN) representation

The approaches developed at Imperial College have at their heart novel process representations which allow to apply several exact optimization methods and decomposition techniques. The first approach was the mathematical programming approach of Kondili et al. [44] based on a discrete representation of time and the newly introduced *state-task network* (STN) representation of the process. The STN representation has three main advantages:

- it distinguishes the process operations from the resources that may be used to execute them, and therefore provides a conceptual platform in which the unique assignment assumption is relaxed and unit-to-task allocation is optimized;
- it avoids the use of task precedence relations which become very complicated in multipurpose plants (a task can be scheduled to start if its input materials are available in the correct quantities and other resources, *e.g.*, processing equipment and utilities, are also available, regardless of the plant history);

- it provides a means of describing very general process recipes, involving batch splitting and mixing and material recycles, as well as storage policies including zero-wait, no-intermediate storage, multipurpose storage tanks and so on.

The formulation of Kondili et al. [44] (described in more detail in Kondili *et al.* [45] is based on the definition of binary variables that indicate whether tasks start in specific units of equipment at the beginning of each time period, together with associated variable batch sizes. Other key variables are the quantity of material in each state held in dedicated storage devices over each time interval, and the quantity of each utility required for processing tasks. Their key constraints are related to equipment and utility usage, material balances and capacity constraints. The common, discrete time grid captures all the plant resource utilizations in a straightforward manner; discontinuities in these are forced to occur at the predefined interval boundaries. Their approach was hindered in its ability to handle large problems by the weakness of the allocation constraints and the general limitations of discrete-time approaches such as the need for relatively large numbers of grid points to represent activities with significantly different durations. Shah et al. [66] modified the model to improve its relaxation properties significantly and therefore increase the scope of applicability considerably.

Pantelides [55] presented a critique of the STN and associated scheduling formulations and argued that despite its advantages, it suffers from a number of drawbacks:

- the model of plant operation is somewhat restricted (each operation is assumed to use exactly one major item of equipment throughout its operation);
- tasks are always assumed to be processing activities which change material states (changeovers or transportation activities have to be treated as special cases);
- each item of equipment is treated as a distinct entity (this introduces solution degeneracy if multiple equivalent items exist);
- different resources (materials, units, utilities) are treated differently, giving rise to many different types of constraints, each of which must be formulated carefully to avoid unnecessarily increasing the integrality gap.

Pantelides [55] then proposed an alternative representation, the *resource-task network* (RTN), based on a uniform description of all resources. In contrast to the STN approach, where a task consumes and produces materials while using equipment and utilities during its execution, in this representation, a task is assumed only to consume and produce resources. Processing items are treated as though consumed at the start of a task and produced at the end. Furthermore, processing equipment in different conditions (e.g., ”clean” or ”dirty”) can be treated as different resources, with different activities (e.g., ”processing” or ”cleaning”) consuming and generating them – this enables a simple representation of changeover activities. He also proposed a discrete-time scheduling formulation based on the RTN which, due to the uniform treatment of resources, only requires the description of three types of constraints, and does not distinguish between identical equipment items.

4.2.2 Decomposition: batching and batch scheduling

The scheduling problem described in Appendix A has also been tackled by the decomposition approach developed by Schwindt and Trautmann [63] and Trautmann [69]. The basic idea of this approach is to decompose the problem into *batching* and *batch scheduling*. In the first step, the number and the size of the batches to be produced is determined. The second step generates a feasible schedule and computes the start and end times of the batches.

The approach considers multi-stage production using multi-purpose equipment. Final products are produced according to a sequence of *tasks*. Production requires reactor time, utilities (such as energy, water, etc.), storage capacities, and possibly personnel. The connections between reactors and storage devices are described by a fixed topology which allows divergent, convergent and cyclic material flows. For further details and underlying assumptions see Neumann et al. [52].

The batching step decomposes the demand into feasible and appropriate batches. The decision variables associated with this step are the number of batches per task, the size of the batches, and the production or conversion rates between input and output product flows. In this step, the objective is to minimize the number of batches weighted by the process times of those batches subject to the constraints:

- the remaining quantity of pre-products not consumed in the production process has to observe the lower and upper storage limits (safety stock, storage capacity);
- the size of the batches has to fulfill the *just-in-time constraints* for those intermediate products for which no storage capacity is available (reactors charge directly to reactors of the next production stage);
- the size of batches and the production or conversion rates have to be within the technological limits.

In [69] and Neumann et al. [52] we find a mixed-integer formulation of the batching problem with nonlinear constraints assuming that for all batches of a task the same size is chosen. The batching problem is solved approximately within seconds.

During the second step, batch scheduling, a feasible schedule of minimal makespan is generated providing the start and end times of all tasks as well as the assignment of resources to the task. The Branch&Bound method presented in Trautmann [69] and Schwindt and Trautmann [63] for solving the batch-scheduling problem, is based on models and methods of resource-constrained project planning (see, for instance, [13] or [62]). The key idea of this method is a decomposition of the problem into *temporal scheduling* and *generation of additional constraints*. Time planning corresponds to the solution of a problem with relaxed resource constraints, i.e., initially it is assumed that all resources are available without any limit. The remaining constraints are time-window constraints originating from production breaks, earliest time for delivering a product, due dates, expiration dates of stored products etc. This temporal scheduling problem corresponds to the determination of the longest path in a graph and can (when appropriately modified to account for forced production breaks and calendar conditions, e.g., respecting holidays) be solved by an extended *label–correcting-algorithm* [69], i.e., computing a schedule whose activities start as early as possible while observing all lower and upper bounds on the differences between termination and starting times is equivalent to compute the longest path from the source to all nodes in a graph [51]. If the plan generated is feasible for the whole problem, the algorithm stops. Otherwise, a point in time is determined at which, now considering the limited resources, a resource constraint would be violated or a batch is started without having uniquely fixed all resources. In the first case, the violation of the resource constraints is eliminated by adding precedence relations between activities; in the second case, the resources are selected. In both cases, the alternative precedence relations or the alternative resources are enumerated by adding them to the search tree. The new subproblem, differing from the previous one by the constraints added, is solved as a temporal scheduling problem. If this problem is not feasible, another subproblem of the tree is chosen. If it is solvable, the scheme of adding additional constraints is continued until a feasible plan is found. The first feasible plan already imposes an upper bound but the search tree might be further explored. In order to restrict the computing time, only a partial search is applied namely to that part of the tree in which one might expect to find good solutions (*filtered beam search*).

Although Trautmann and Schwindt only consider the makespan objective, their approach is open to account for other objective functions, e.g., to minimize the delay with respect to some due dates. Another advantage of their approach is that re-scheduling is supported, for instance, if some resources suddenly fail.

4.2.3 Special algorithms for problems with time-precedence and aggregate resource constraints

Some algorithms applied successfully to the problem described in Section 3.3 are based on the original work by Bartusch et al. [5]. The basic idea is to permit temporal constraints in the form of arbitrary, context-sensitive time lags between start and/or completion times of activities and time-dependent resource requirements and availabilities in the form of piece-wise constant step functions. The B&B algorithm constructs feasible schedules along the time axis, starting at time 0 and, at every decision time t, branches into feasible decisions (about which set of jobs to start at t) with respect to the resource constraints, while always observing the temporal constraints. It thus always maintains a set of best possible feasible partial schedules for the different subproblems created by these decisions and seeks to either complete them into a full schedule, or to discard a branch because the best value that can be obtained by this branch exceeds the current best objective value.

This approach is combined with heuristics to generate good feasible solutions in order to start with good upper bounds, different strategies for exploring the subproblems created (best-first-search, depth-first-search, user-defined search), and several methods for generating good lower bounds for best possible value of the subproblems. In addition, it uses a powerful domination rule which permits discarding a subproblem if the currently "active" jobs in the partial schedule have been encountered already in a different branch with a better lower bound. This domination rule is based on an efficient search tree implementation of all sets of active jobs encountered so far. The most recent progress and results have been achieved by the group around R. Möhring at TU Berlin and are based on Lagrangian Relaxation and Branch&Bound techniques.

Lagrangian relaxation: In order to compute lower bounds on the optimal objective function value for resource-constrained project scheduling problems, an approach via Lagrangian relaxation has been suggested [49]. The Lagrangian relaxation is based on a well-known time-indexed integer linear programming formulation of the problem [57]. The same Lagrangian relaxation has been used before [18]. The basic idea is to relax the resource-constraints, and to penalize their violation in the usual Lagrangian fashion. This results in a Lagrangian subproblem which is a so-called project scheduling problem with start-time dependent costs. In this problem, each job incurs a cost which depends on its start time, the jobs are subject to precedence constraints (or arbitrary time lags), and the objective is to find a schedule which has minimal costs. This problem, or special cases thereof, has been addressed frequently in the literature. [50] give an overview of these results. As it turns out, the project scheduling problem with start-time dependent costs can be efficiently solved as a minimum cut problem in a directed graph. This insight is the key to the practical efficiency of the Lagrangian approach, which uses a sub-gradient method to iterate the Lagrangian multipliers and provides reasonable strong lower bounds within very moderate computation times. Note that the lower bounds are theoretically the same as those obtained by the LP-relaxation of the problem, but in the current case the Lagrangian relaxation method works much faster than Simplex type algorithms or interior point methods. In addition, [49] propose to exploit the dual information from solutions of the Lagrangian relaxation in order to compute also feasible solutions to resource-constrained project scheduling problems. The basic idea is borrowed from previous work on approximation algorithms for machine scheduling problems. In this context, it was shown by various authors that LP-based list-scheduling, combined with the concept of so-called α-completion times, can lead to schedules with constant worst-case performance guarantees. The rationale behind this approach is that a solution of a relaxed problem also holds valuable information to compute feasible solutions to the original, resource-constrained problem. [49] use simple list scheduling algorithms which are based on priority lists according to α-completion times of jobs in the solutions of the Lagrangian subproblems. Feasible solutions are computed in each iteration of the subgradient optimization. The computational experiments with this approach are very promising in terms of both computation time and solution quality. In fact, on a set of well established benchmark instances, the Lagrangian based approach provides solutions which are comparable to those of state-of-the-art algorithms from the literature. Moreover, for instances close to real-world scenarios, the results are clearly favorable to those obtained with a constraint propagation approach by [34], or a branch-and-bound algorithm by [27]. Note that the Lagrangian approach can handle many regular, and also non-regular objective functions.

Branch-and-Bound: For resource-constrained problems with arbitrary minimal and maximal time lags, also called *time windows* or *generalized precedence constraints*, it is already NP-hard to compute a *feasible* schedule. Many order theoretic insights into the structure of optimal solutions have been already obtained by [5]. Partially based on ideas of their work, branch-and-bound algorithms have been proposed and evaluated more recently by [19], [62], and [27]. The underlying idea of the algorithms is that *time-feasible schedules* are enumerated by systematically resolving *resource conflicts*. Here, a time-feasible schedule denotes a schedule which does not violate the time lag constraints, and a resource conflict is a time phase during which the schedule violates the resource constraints. The resource conflicts are resolved by introducing additional precedence relations between jobs, or sets of jobs ([19], [62]), a concept which is based on an order theoretic representation theorem of optimal schedules (see Theorem 3.8 in [5]). In contrast, [27] resolve resource conflicts by a dynamic update of release dates instead of introducing precedence relations. Thus, their algorithm is not based on the order theoretic concept described in [5], but on a very simple dominance property instead. At a first glance, this technique has the drawback that resource conflicts are resolved only locally. Nevertheless, subject to several additional features which help to truncate large parts of the enumeration tree, the computational results show that the algorithm performs better than previous algorithms which are based on the idea to resolve resource conflicts. Compared to the previous branch-and-bound approaches, the efficiency is partly due to the efficient update of the time-feasible schedules in each node of the enumeration tree. Other branch-and-bound algorithms by [34], [22], and [23] are based on constraint propagation. These algorithms rely on the idea to reduce the possible start times of jobs as much as possible by propagating corresponding lower and upper bounds in every node of the enumeration tree. The results with these algorithms are also very good. As a rule of thumb, however, for large-scale instances, the computation times of all available branch-and-bound implementations are prohibitive. In these cases the algorithms are generally used as heuristics by only evaluating parts of the enumeration tree.

4.2.4 Heuristics and meta-heuristics

Besides the exact methods described so far, a variety of heuristics, according to Glover [29] better called *meta-heuristics*, are used to solve scheduling problems [21] by simulating a given system and evaluating its *function of merit* (the objective function in exact optimization). Some well known meta-heuristics, i.e., techniques which are not problem specific and are based on generic principles and schemes which can be used to construct problem-specific heuristics are: *genetic algorithms* (GA), *simulated annealing* (SA) and *tabu search* (TS). All meta-heuristics have in common that they usually lack the proof of convergence and the proof of optimality. However, they can be effectively used to improve a given solution by performing a local search coupled with some exchange mechanisms based on appropriate neighborhood relations. SA and TS have the advantage that they can leave *poor* local optima and move to *better* solutions.

SA (see, for instance, [1], [25] or [2]) links the probability of accepting a solution which is worse than the reference solution to a temperature-like parameter which describes the cooling of metals. This approach which introduces a non-deterministic argument for accepting an inferior solution is the key-element to leave *poor* local optima.

TS [29] is a meta-strategy for guiding known heuristics past the traps of local optimality. It exploits knowledge from previous solutions and thus uses an abstract memory. Popularized by Glover in the early 90s, TS has been applied to integer programming problems involving scheduling, routing, traveling salesman and related problems.

GA ([35], [36]) – an algorithm in the class of evolutionary algorithms – uses population of solutions subject to survival of the fittest criteria, mutation and recombination of positive properties in very solutions.

5 Conclusions and summary

We have provided an overview of planning and scheduling in the process industry. The state-of-the-art technology based on mathematical, especially mixed-integer optimization for planning is quite advanced and appropriate for solving real world planning problems. Mixed integer optimization can provide a quantitative basis for decisions and allow to cope most successfully with complex problems and it has proven itself as a useful technique to reduce costs and to support other objectives. Despite that, this technology has not yet found its way into many commercial software packages. For scheduling problems, there is not yet a commonly accepted state-of-the-art technology although some promising approaches have been developed, especially for job shop problems. Nevertheless, the majority of software packages is still based on pure heuristics.

What will the future hold for us in planning and scheduling? There is a growing number of software packages available which support – at least to some extent – the application of exact methods for a variety of planning problems. That way, planning based on mathematical, especially mixed-integer optimization becomes more and more the state-of-the art in the chemical, food and pharmaceutical industry and as well in refineries. For scheduling this is not yet the case. Most approaches are still based on heuristics. But there is light at the end of the tunnel: hybrid approaches in which MILP, constraint programming and graph theory are used together are in the process of being developed, and may be a common language for scheduling problems.

There is a trend over the last few years bringing mathematical programming and the constraint programming community closer to each other. This results in hybrid approaches (see, for instance, [32] and [37]), i.e., in a language and algorithms combining elements from both communities. This may have a great impact on supply chain problems and scheduling. In 1999, the European Commission awarded the project LISCOS (Large Integrated Supply Chain Optimization Software) with several million Euros. The technical core of this project initiated by BASF's mathematical consultant group and 8 other partners is the development of MIP-CP hybrid techniques (http://www.liscos.fc.ul.pt). Timpe [68] describes a successful application of this techniques to a real world planning and scheduling problem in the process industry.

Another focus of modeling which is possible now due to increased computer power available is the opportunity to solve design and operational planning problems, or strategic and operational planning problems simultaneously in one model. The motivation, ideas how to to this and successful examples are provided by [40].

Finally, it is now also possible to give up the assumption that all data have to be treated as determinstic data – note that in this article we focussed only on deterministic models. However, as some data, e.g., demand forecast in planning models, or production data in scheduling may be subject to uncertainties, is seems to be advantegous if we could give up the assumption that planning and modeling is exclusively based on deterministic data. In that case, stochastic optimization is the mean of choice. Nowadays, there exist powerful solution techniques to combine mixed-integer programming and stochastic optimization (see, for instance, Schultz [61] or Carøe and Schultz [16]) when data are subject to uncertainties. Successful applications of this techniques to scheduling problems in the chemical process industry are reported, for instance, by Sand et al. [59] or Engell et al. [26]. Alternatively, to these techniques there exists an approach [6] to find robust solutions to LP-problems including uncertain data.

While MIP has already well established itself in planning, further quantum leaps in scheduling are to be expected from the combination of several techniques such as mathematical optimization, graph theory and constraint programming, and to exploit problem specific structures. Exact, or at least much better scheduling techniques knock at our doors and might, say, within 5 to 10 years, play a similar role as does MIP in planning nowadays.

Appendix: A benchmark problem (Westenberger and Kallrath)

Due to the increasing interest in the benchmark scheduling problem formulated by Westenberger and Kallrath (1994) in a joint internal publication by Bayer AG (Leverkusen) and BASF Aktiengesellschaft (Ludwigshafen), this appendix summarizes it and thus makes it available to a broader audience. The case study covers most of the features that contribute to the complexity of batch process scheduling in industry. The problem is to some extent presented in the language of an MILP model. This is to guarantee uniqueness in the interpretation. Nevertheless, people interested in the problem should be encouraged to try other approaches, e.g., time-continuous formulations. A state-task-network representation of the production process and the possible assignment of tasks to production units are given in Figure 3 in [10].

1 Description of the problem

The production process considered consists of a network of 17 processing tasks, 19 states, 9 production units, and 37 divergent, convergent and cyclic material flows. Moreover, the production process includes flexible proportions of output goods (see Task 2/2), cyclical material flows (recycling of output from Task 3/3 into State 1), and several intermediates (see state nodes 5, 9, 10, and 12), which do not allow storage between processing steps. All processing tasks are performed in batch mode with lower and upper bounds on the batch sizes. These bounds are pre-determined from properties of the processes and the capacities of the reactors. Batch sizes are treated as decision variables and may be different even for the same type of product or the same production unit.

Table 1. Description of production units

Production unit	Number of processing lines	Number of products	maximum batch size	minimum batch size
R_u^P	L_u	P_u / unit	A_u^+ / kg	A_u^- / kg
R_1	1	1	10	3
R_2	1	2	20	5
R_3	1	2	10	4
R_4	1	4	10	4
R_5	1	2	10	4
R_6	2	3	7	3
R_7	2	5	12	4

In this production planning problem a variety of products are produced by a process plant consisting of a number U of production units $R_u^P, u = 1, \ldots, U$. We suppose an one-to-one relation of process steps and production units in the plant. Every production unit R_u^P contains L_u *production lines* R_{ul}^L, $l = 1, \ldots, L_u$. In the current benchmark example we have $U = 7$ and the vector L_u specified in Table 1. Each production unit R_u^L generates a number P_u of *products* P_{up}, $p = 1, \ldots, P_u$. The number of products per production unit is given in Table 1 as well.

Remark. The production is realized in batch mode, i.e., certain quantities of ingredients are processed by a given *process time per batch process* T_{ulp}. Process times depend on the production line R_{ul}^L used for processing and the product to be made (see Table 2), but in our case, not on actual batch sizes.

In our example we have at most two different ingredients per batch (see product P_{73}) and at most 2 generated products per batch. For instance, productions of P_{21} and P_{22} are coupled. The same holds for P_{31} and P_{32}.

Remark. Note that F_{211n}^{o1} is restricted by $0.2 \leq F_{211n}^{o1} \leq 0.7$ and that F_{311n}^{o1} is explained in Section 5 of this appendix. Only those processing lines are allowed to make product P_{up} which have a process time T_{ulp} for this product declared in Table 2. For example, production of P_{73} is allowed to run on R_{71}^L, but not to run on R_{72}^L.

We refer to batch processes as B_{ulpn}, where B_{ulpn} is a pointer to the n^{th} batch process P_{up} running on production R_{ul}^L. The integer variable $\nu_{ulp} \in \mathbb{N}_0$ describes the total number of batch processes to be performed for product P_{up} in production line R_{ul}^L. In case of two products we declare the first one as the main product and the second one as a side product. The side product can be interpreted as a less attractive product or as a waste product. Each batch has a *specific batch size* described by the variable a_{ulpn} subject to a maximum size A_u^+ and a minimum size A_u^-. Values for A_u^+ and A_u^- are given in Table 1.

The fraction of the k^{th} ingredient of a batch B_{ulpn} is specified by F_{ulpn}^{ik}. The fraction of k^{th} output is given by F_{ulpn}^{ok}. Because of a maximum of two ingredients and a maximum of two output components, it is sufficient to specify the fraction of the first ingredient and the fraction of the main product only.

Table 2. Description of batch processes

P_{up}	$P_{up'}$	F^{o1}_{ulpn}	r^P_u	R^L_{ul}	T_{ulp}	P_{vr}	$P_{v'r'}$	F^{i1}_{ulpn}
P_{11}	–	1.	R_1	R^L_{11}	0.05 d	P_0	–	1.
P_{21}	P_{22}	F^{o1*}_{211n}	R_2	R^L_{21}	0.1 d	P_{11}	–	1.
P_{31}	P_{32}	F^{o1**}_{311n}	R_3	R^L_{31}	0.05 d	P_{22}	–	1.
P_{41}	–	1.	R_4	R^L_{41}	0.1 d	P_{21}	–	1.
P_{42}	–	1.	R_4	R^L_{41}	0.1 d	P_{21}	–	1.
P_{43}	–	1.	R_4	R^L_{41}	0.1 d	P_{31}	–	1.
P_{44}	–	1.	R_4	R^L_{41}	0.1 d	P_{31}	–	1.
P_{51}	–	1.	R_5	R^L_{51}	0.15 d	P_{21}	–	1.
P_{52}	–	1.	R_5	R^L_{51}	0.15 d	P_{31}	–	1.
P_{61}	–	1.	R_6	R^L_{61}	0.100 d	P_{42}	–	1.
P_{62}	–	1.	R_6	R^L_{61}	0.125 d	P_{43}	–	1.
P_{63}	–	1.	R_6	R^L_{61}	0.150 d	P_{44}	–	1.
P_{61}	–	1.	R_6	R^L_{62}	0.125 d	P_{42}	–	1.
P_{62}	–	1.	R_6	R^L_{62}	0.150 d	P_{43}	–	1.
P_{63}	–	1.	R_6	R^L_{62}	0.150 d	P_{44}	–	1.
P_{71}	–	1.	R_7	R^L_{71}	0.100 d	P_{51}	–	1.
P_{72}	–	1.	R_7	R^L_{71}	0.100 d	P_{52}	–	1.
P_{73}	–	1.	R_7	R^L_{71}	0.100 d	P_{41}	P_{61}	1/2
P_{74}	–	1.	R_7	R^L_{71}	0.150 d	P_{62}	–	1.
P_{75}	–	1.	R_7	R^L_{71}	0.150 d	P_{63}	–	1.
P_{71}	–	1.	R_7	R^L_{72}	0.150.d	P_{51}	–	1.
P_{72}	–	1.	R_7	R^L_{72}	0.150 d	P_{52}	–	1.
P_{74}	–	1.	R_7	R^L_{72}	0.150 d	P_{62}	–	1.
P_{75}	–	1.	R_7	R^L_{72}	0.150 d	P_{63}	–	1.

The columns in this table have the following meaning:
main and side product, fraction of main product, production unit
process time per batch, ingredient 1 and 2, fraction of ingredient 1

If we introduce the variables m^{ik}_{ulpn} and m^{ok}_{ulpn} with

$$m^{i1}_{ulpn} = F^{i1}_{ulpn} a_{ulpn} \quad , \quad m^{i2}_{ulpn} = (1 - F^{i1}_{ulpn}) a_{ulpn} \tag{6}$$

$$m^{o1}_{ulpn} = F^{o1}_{ulpn} a_{ulpn} \quad , \quad m^{o2}_{ulpn} = (1 - F^{o1}_{ulpn}) a_{ulpn} \tag{7}$$

to describe the quantities of ingredients and output of products in a batch process B_{ulpn}, the mass conservation reads

$$m^{i1}_{ulpn} + m^{i2}_{ulpn} = a_{ulpn} = m^{o1}_{ulpn} + m^{o2}_{ulpn} \tag{8}$$

Each batch B_{ulpn} has a unique *start time* t^S_{ulpn} and a unique *end time* t^E_{ulpn} related by

$$t^E_{ulpn} = t^S_{ulpn} + T_{ulp} \quad , \quad \forall \{ulpn\} \tag{9}$$

Remark. For the sake of simplicity we assume that process time T_{ulp} does not depend on the actual batch size of any batch job B_{ulpn}.

Call a batch $B_{u'l'p'n'}$ the *successor* to batch B_{ulpn} if both batches have to run on the same production line and if there is no other job in between them running on the same processing line. We can express this condition mathematically.

A batch job $B_{u'l'p'n'}$ is called *successor* to batch B_{ulpn} if $u = u'$, $l = l'$ and $t^S_{ulpn} \subset t^S_{ulp'n'}$ is true and for each $B_{ulp''n''}$ with $p'' \subset P_u + 1$ and $n'' \subset N_{ulp''} + 1$ one of two relations (10) or (11)

$$\left\{(t^S_{ulp''n''} \leq t^S_{ulpn}) \wedge (t^S_{ulp''n''} \leq t^S_{ulp'n'})\right\} \tag{10}$$

$$\left\{(t^S_{ulp''n''} \geq t^S_{ulpn}) \wedge (t^S_{ulp''n''} \geq t^S_{ulp'n'})\right\} \tag{11}$$

is true. In an MILP model we could use an auxiliary binary variable $\alpha_{ulpnp'n'}$

$$\alpha_{ulpnp\prime n\prime} = \begin{cases} \text{1, if batch } B_{u'l'p'n'} \text{ is the successor to batch } B_{ulpn} \\ \text{0, otherwise} \end{cases}$$

to indicate a predecessor-successor relation of two batches B_{ulpn} and $B_{u'l'o'n'}$ and to describe this feature.

2 Cleaning processes

Sometimes a batch process requires a cleaning process C^P_{ulpn}. We define a slack variable $\eta_{ulpn'} \geq 0, n' \leq N_{ulp}$ measuring the machine time of production line R^L_{ul} that is not used for production or cleaning processes between a batch B_{ulpn} and its successor. The auxiliary binary variable β_{ulpn} is a decision variable with

$$\beta_{ulpn} := \begin{cases} \text{1, if a batch } B_{ulpn} \text{ is followed by a cleaning process} \\ \text{0, otherwise} \end{cases} \tag{12}$$

For all $\alpha_{ulpnp'n'} = 1$ we require that

$$e_{ulpn} + \beta_{ulpn}(F^C T_{ulp} + \eta_{ulpn}) = t^S_{ulp'n'} \tag{13}$$

where F^C is a factor used to describe the cleaning time: $F^C T_{ulp}$.

Two conditions enforce cleaning processes ($\beta_{ulpn} = 1$): (*i*) if $\alpha_{ulpnu'n'} = 1$ and $p \subset p'$ then a cleaning process has to be included. (*ii*) if a batch is the last batch process on a processing line a cleaning process has to follow.

Remark 1. A sequence of products with decreasing quality requirements can help to avoid cleaning processes. This viewpoint motivates condition (*i*) in our example. For this reason a schedule should prefer batch sequences $p \geq p'$ for predecessor-successor-pairs ($\alpha_{ulpnp'n'} = 1$).

Remark 2. Normally, cleaning time depends on the specific batch product and production line. For the sake of simplicity we assume a cleaning time that depends only on the duration of the predecessor batch and on a given factor F^C.

Remark 3. Production lines which are not in use should be cleaned to avoid ongoing reactions of residues. Equation (13) models this requirements because (13) can only be fulfilled if a positive slack time is combined with a cleaning process.

Remark 4. Different batch processes cannot run on the same production line simultaneously, i.e., $t^E_{ulpn} \leq t^S_{ulp'n'}$ for all $(u, l, p, n, p', n' | \alpha_{ulpnp'n'} = 1)$.

3 Stock conditions and plant input/output

The storable products P_{up} are given by the index set $(u, p) \in \mathcal{S}$ with

$$\mathcal{S} = \{(1,1), (2,2), (3,1), (4,2), (4,3), (4,4), (6,1), 6,3)\} \tag{14}$$

The products P_{41}, P_{51}, P_{52}, P_{62} cannot be stored.

The quantity of product P_{up} stored at time t is described by the stock function $j_{up}(t)$. $j_{up}(t)$ is completely defined by the initial value $j_{up}(0)$ and the cumulative mass inflow $q^i_{up}(t)$ and outflow $q^o_{up}(t)$ following from the schedule of all batches $B_{u'l'p'n'}$ of all products:

$$q^i_{up}(t) = f(t, s_{ulpn}) \quad , \quad q^o_{up}(t) = f(t, e_{ulpn}) \tag{15}$$

with

$$f(t, t_r) := \sum_{l=1}^{L_u} \sum_{n=1}^{N_{ulp}} H(t, t_r) a_{ulpn}, \quad H(t, t_r) = \begin{cases} 1, \text{ if } t \geq T \\ 0, \text{ else} \end{cases} \tag{16}$$

$j_0(t)$ describes the time evolution of raw material requirement, *i.e.*, $j_0(t) = q^i_{11}(t)$. The product outflow j_{7p} of end products is given by $j_{7p}(t) = q^o_{7p}(t)$ for all $p = P_1, \ldots, P_7$.

4 Stock balance equations

The current quantity of a certain product in stock is given by its initial quantity and the cumulative mass inflow and outflow associated with the production of that product. If we explicitly resolve the production scheme we get

$$j_{11}(t) = j_{11}(0) + q^o_{11}(t) - q^i_{21}(t) + \sum_{n=1}^{N_{311}} (1 - F^{o1}_{311n}) a_{311n} H_{e_{311n}}(t) \tag{17}$$

$j_{11}(t)$ takes into account the re-usage of P_{32} as product P_{11}

$$j_{21}(t) = j_{21}(0) + \sum_{n=1}^{N_{211}} F^{o1}_{211n} a_{211n} H_{e_{211n}}(t) - q^i_{41}(t) - q^i_{42}(t) - q^i_{51}(t) \tag{18}$$

$$j_{22}(t) = j_{22}(0) + \sum_{n=1}^{N_{211}} (1 - F^{o1}_{211n}) a_{211n} H_{e_{211n}}(t) - q^{i}_{31}(t) \tag{19}$$

$j_{21}(t) = j_{22}(t)$ share the output of R_2

$$j_{31}(t) = j_{31}(0) + \sum_{n=1}^{N_{311}} F^{o1}_{311n} a_{311n} H_{e_{311}}(t) - q^{i}_{43}(t) - q^{i}_{44}(t) - q^{i}_{52}(t) \tag{20}$$

In j_{31} the stock input is reduced by the separated quantity of P_{32} (see j_{11}).

$$j_{4k}(t) = j_{4k}(0) + q^{o}_{4k}(t) - q^{i}_{6,k-1}(t) \quad , \quad k = 2, 3, 4$$

$$j_{61}(t) = j_{61}(0) + q^{o}_{61}(t) - \sum_{l=1}^{L_7} \sum_{n=1}^{N_{713}} (1 - F^{i1}_{713n}) a_{713n} H_{s_{713n}}(t)$$

$$j_{63}(t) = j_{63}(0) + q^{o}_{63}(t) - q^{i}_{75}(t)$$

The products which are not storable, lead to the equality constraints

$$q^{o}_{51}(t) = q^{i}_{71}(t), \quad q^{o}_{52}(t) = q^{i}_{72}(t) q^{o}_{62}(t) = q^{i}_{74}(t), \quad q^{o}_{41}(t) = F^{i1}_{713n} q^{i}_{73}(t)$$

The last equation uses the fact that F^{i1}_{713n} is a constant and does not depend on n.

5 Objective functions

Now, several tasks, i.e., optimization problems, can be formulated using the definitions provided in Section 5.

5.1 Task 1: Minimize makespan – compute a pre-schedule. The makespan to be minimized corresponds to the latest completion time of any of the processing tasks or cleaning operation. The rationale for this objective function is that the plant may be reconfigured at the earliest possible time. The demand D^{A}_{7p} to be satisfied in time period t^{T_1}, is given in Table 3. At time t^{T1} the stock function $j_{7p}(t^{T1})$ has to be equal or greater than the demand D^{A}_{7p}, $o = 1, \ldots, 5$. In Task 1, we set $F^{o1}_{311n} = 1$ for all n, i.e., the re-usage of P_{32} as P_{11} is not considered. Furthermore, cleaning processes $(F^{C} = 0)$ are neglected.

Bounds, $J^{-}_{up} \leq j_{up}(t) \leq J^{+}_{up}$, on stock and initial stock conditions for all storable products $(u, p) \in \mathcal{S}$ (see Section 3 of this appendix) are given in Table 4.

Task. Compute the smallest value t^{T_1} for feasible batch plans consistent with all previous constraints!

Table 3. Product demand

j	1	2	3	4	5
D^A_{7j}/kg	0	0	90	50	40
D^B_{7j}/kg	30	30	40	20	40

Table 4. Stock conditions ($J^-_{up} = 0$ for all)

[kg]	j_{11}	j_{21}	j_{22}	j_{31}	j_{42}	j_{43}	j_{44}	j_{61}	j_{63}
$j_{up}(0)$	20	20	0	20	0	0	0	0	0
J^+_{up}	30	30	15	30	10	10	10	10	10

5.2 Task 2: Task 1 considering cleaning times. In addition to Task 1, the cleaning processes described in Section 2 of this appendix should be considered with F^C being set to $F^C = 0.5$. Determine t^{T_2} as small as possible to satisfy the demands A specified in Table 3.

5.3 Task 3: Task 1 consideration of a cyclic product flow. In addition to Task 1, take into account re-usage of product P_{32} as product P_{11} as described in Section 1 of this appendix and thereof set $F^{o1}_{311n} = 0.6923$ for all n. Neglect cleaning processes $(F^C = 0)$ and minimize $t^{T_{23}}$.

5.4 Task 4: Minimize makespan – compute a complete schedule. Take into account cleaning processes and a cycle in product flow and set $F^C = 0.5$ and $F^{o1}_{311n} = 0.6923$ for all n. The production requirement D^B_{7p} specified in Table 3 has to be fulfilled exactly $j_{7p}(t^{T_4}) = D^B_{7p}$ for all $p = 1, \ldots, 5$. Stock limitations are the same as in Task 1, but unlike Task 1, here, initial stock values $j_{up}(0)$ are not prescribed.

Task. Minimize t^{T_4} so that $j_{up}(0) = j_{up}(t^{T4})$ for all $(u,p) \in \mathcal{S}$ with $u \subset 7$! All batch processes and cleaning processes should be finished at cycle time t^{T_4}.

5.5 Task 5: Maximize profit. Take the parameter values as in Task 4 (cyclic production, requirement of Table 5 etc.). Here, a time interval of 2 days is defined ($t^{T_5} = 2d$). A product requirement which has to be satisfied is not prescribed. Instead, market prices $C^p(P_{up})$ in units of \$/kg for raw material P_0 and end products P_{71}, P_{72}, P_{73}, P_{74}, P_{75} are given as $[5, 10, 10, 30, 20, 15]$.

Task. Find a product mix and a cyclic schedule which maximize the profit function w

$$w = -C^p(P_0)j_0(t^c) + \sum_{p=1}^{5} C^p(P_{70})j_{70}(t^c) \qquad (21)$$

5.6 Task 6: Minimal cost capacity design. Take parameter values as in Task 4. Again, product requirement is given by Table 5. Find appropriate stock capacities

J_{up}^{+} and appropriate production rates r_{ul}^{P} for each production line l. Note that the stock capacities J_{up}^{+} are variable now.

Here, the production rate of a production line should be the same for each product manufactured on this line, i.e., the batch process time T_{ulp} does not depend on product index p. The batch process time should now be redefined by the maximum batch size A_{u}^{+} and the production rate r_{ul}^{P}, i.e., $T_{ulp} = A_{u}^{+} r_{ul}^{P}$. For the sake of simplicity we assume the same investment cost for all production rates (C^{R} =10 $/kg/d) and the same investment cost of stock capacity for all products ($C^{S} = 3$ $/kg).

Task. Determine optimal values for all r_{ul}^{P} and J_{up}^{+} so that demand specified in Table 5 under the constraints of Task 4 can be satisfied in time intervals of length $t^{T_6} = 0.5$ days and such that the investment cost c

$$c = \sum_{u=1}^{7} \sum_{l=1}^{L_u} C^{R} r_{ul}^{P} + \sum_{(u,p)\in\mathcal{S}} C^{S} J_{up}^{+} \tag{22}$$

are minimized, where $\mathcal{S}$ is the set of all indexes (u,p) of all storable products P_{up} defined in Section 3 of this appendix.

6 List of symbols

Symbol	*Description*
$l = 1,\ldots,L_u$	index referring to processing lines
$n = 1,\ldots,N_{ulp}$	index referring to batch processes
$p = 1,\ldots,P_u$	index referring to products
$u = 1,\ldots,U$	index referring to processing units
A_{u}^{-}, A_{u}^{+}	minimum and maximum batch size (fixed)
a_{ulpn}	batch size (variable)
B_{ulpn}	pointer to the n^{th} batch of product P_{up} running on production line R_{ul}^{L}
$C^{p}(P_{up})$	product P_{up} (fixed)
C^{r}	investment cost for additional production capacity
C^{S}	investment cost for additional stock capacity
D_{Up}	demand for product p
F_{ulpn}^{i1} (variable)	fraction of first ingredient P_{vr} of batch B_{ulpn}
F_{ulpn}^{o1} (variable)	fraction of first output product P_{up} of batch B_{ulpn}
$F^{C} \geq 0$	factor describing the cleaning time: $F^{C} T_{ulp}$ (fixed)
$H_{R}(t)$	Heavyside function
J_{up}^{-}	minimum and maximum stock of product P_{up}
$j_{up}(t)$ (variable)	stock as a function of time t
$j_{0}(t)$	cumulative requirement of raw material
$m_{ulpn1}^{i}, m_{ulpn2}^{i}$	quantities of ingredients of batch B_{ulpn} (variable)

Symbol	*Description*
m^o_{ulpn1}, m^o_{ulpn2}	quantities of products of batch B_{ulpn} (variable)
N_{ulp} (variable)	number of batches of product P_{up} active on line R^L_{ul}
$P_{up}, p = 1, \dots, P$	product p made in production unit R_u
$q^o_{up}(t)$, $q^o_{up}(t)$	cumulative mass inflow and outflow (variable)
$R_u, u = 1, \dots, U$	production unit related to a specific process step
$R^L_{ul}, l = 1, \dots, L$	processing line (part of production unit R_u)
r^P_{ul}	production rates defined in Task 6 (variable)
$\mathcal{S}$	index set of storable products (fixed)
t^E_{ulpn}, t^S_{ulpn}	end and start time of batches (variable)
T_{ulp}	process time per batch process (variable in Task 6)
$\alpha_{ulpnp'n'}$	binary variable to indicate a predecessor-successor relation of two batches B_{ulpn} and $B_{ulp'n'}$
β_{ulpn}	auxiliary binary decision variable with $\beta_{ulpn} = 1$, if a batch B_{ulpn} is followed by a cleaning process
$\eta_{ulpn} \geq 0, n \leq N_{ulp}$	slack variable measuring the machine time unused for production between a batch B_{ulpn} and its successor.

Remark. The values of variables have to be positive or zero. Variables are usually defined by small letters; small Greek characters denote discrete variables. Exceptions: The maximum stock capacity, J^+_{up}, is a variable in Task 6. F^{o1}_{ulpn} is a variable for special values of (u, l, p, n) (see Table 2).

References

1. Aarts EHL, Korst JHM (1989) Simulated Annealing and Boltzmann Machines. Wiley, Chichester
2. Aarts EHL, Korst JHM, van Laarhoven PJM (1997) Simulated annealing. In: Aarts EHL, Lenstra JK (eds) Local search in combinatorial optimization, pp 91–120. Wiley, Chichester
3. Ahleff J (1995) Analyse von mathematischen Modellen zur Reihenfolgeplanung in der Prozessindustrie. Master thesis, RWTH Aachen, Aachen
4. Applequist G, Samikoglu O, Pekny J, Reklaitis GV (1997) Issues in the use, design and evolution of process scheduling and planning systems. ISA Trans 36: 81–121
5. Bartusch M, Möhring RH, Radermacher FJ (1988) Scheduling project networks with resource constraints and time windows. Annals of Opereration Research 16: 201–240
6. Ben-Tal A, Nemirovski A (2000) Robust solutions of linear programming problems contaminated with uncertain data. Mathematical Programming 88: 411–424
7. Berning G, Brandenburg M, Gürsoy K, Mehta V, Tölle FJ (2002) An integrated system solution for supply chain optimization in the chemical process industry. OR Spectrum 24, No. 4 (to appear)
8. Blömer F (1999) Produktionsplanung und -steuerung in der chemischen Industrie – Ressourceneinsatzplanung von Batchprozessen auf Mehrzweckanlagen. Gabler, Wiesbaden

9. Blömer F, Günther HO (1998) Scheduling of a multi-product batch process in the chemical industry. Computers in Industry 36: 245–259
10. Blömer F, Günther HO (2000) LP-based heuristics for scheduling chemical batch processes. International Journal of Production Research 35: 1029–1051
11. Borodin A, El-Yaniv R (1998) Online computation and competitive analyses. Cambridge University Press, Cambridge
12. Brockmüller B, Wolsey L (1995) BASF Problem 1 modelling – Production scheduling with batch-sizing and storage restrictions. Report of the ESPRIT project PAMIPS, Task 1.4 8/1995, BASF-AG, ZX/ZC-C13, D-67056 Ludwigshafen
13. Brucker P, Drexl A, Möhring M, Neumann K, Pesch E (1999) Resource-constrained project scheduling: Notation, classification, models and methods. European Journal of Operational Research 112: 3–41
14. Burkard RE, Hujter M, Klinz B, Rudolf R, Wennink M (2000) A process scheduling problem arising from chemical production planning. Optimization Methods & Software 10: 175–196
15. Burkard RE, Kocher M, Rudolf R (2000) Rounding strategies for mixed integer programs arising from chemical production planning. Yugoslav Journal of Operations Research 8: 9–23
16. Carøe CC, Schultz R (1999) Dual decomposition in stochastic integer programming. Operations Research Letters 24: 37–45
17. Cerda J, Gutierrez MVJM, Espulgas S, Mata J (1989) A new methodology for the optimal design and production schedule of multipurpose batch plants. Industrial Engeneering Chemical Research 28: 988–998
18. Christofides N, Alvarez-Valdes R, Tamarit JM (1987) Project scheduling with resource constraints: A branch and bound approach. European Journal of Operational Research 29: 262–273
19. DeReyck R, Herroelen W (1998) A branch and bound procedure for the resource-constrained project scheduling problem with generalized precedence relations. European Journal of Operational Research 111(1): 152–174
20. Domschke W, Scholl A, Voß S (1997) Produktionsplanung, 2nd ed. Springer, Berlin Heidelberg New York
21. Dorndorf U, Pesch E (1992) Genetic algorithms for job scheduling. In: Hansmann K-W (ed) Parallel problem solving from nature, 2. Operations Research Proceedings 1992, pp 243–250. Springer, Berlin Heidelberg New York
22. Dorndorf U, Pesch E, Phan Huy T (2000) A time-oriented branch-and-bound algorithm for resource-constrained project scheduling with generalized precedence constraints. Management Science 46: 1365–1384
23. Dorndorf U, Pesch E, Phan Huy T (2000) A time-oriented branch-and-bound algorithm for resource-constrained project scheduling. Mathematical Methods of Operations Research 52: 413–439
24. Drexl A, Haase K (1995) Proportional lotsizing and scheduling. International Journal of Production Economics 40: 73–87
25. Eglese RW (1990) Simulated annealing: A tool for operational research. European Journal of Operational Research 46: 271–281
26. Engell S, Märkert A, Sand G, Schultz R, Schulz C (2001) Online scheduling of multiproduct batch plants under uncertainty. In: Grötschel M, Krumke SO, Rambau J (eds) Online optimization of large scale systems, pp 649–676. Springer, Berlin Heidelberg New York
27. Fest A, Möhring RH, Stork F, Uetz M (1998) Resource constrained project scheduling with time windows: A branching scheme based on dynamic release dates. Technical Report 596/1998 (revised 1999), Fachbereich Mathematik, Technische Universität Berlin, Berlin

28. Fieldhouse M (1993) The pooling problem. In: Ciriani T, Leachman RC (eds) Optimization in industry: mathematical programming and modeling techniques in practice, pp 223–230. Wiley, Chichester
29. Glover F, Laguna M (1997) Tabu search. Kluwer Academic Publisher, Dordrecht
30. Grossmann IE, Caballero JA, Yeomans H (1999) Mathematical programming approaches for the synthesis of chemical process systems. Korean Journal Chemical Engeneering 68: 407–426
31. Grunow M, Günther H-O, Lehmann M (2002) Campaign planning for multi-stage batch processes in the chemical industry. OR Spectrum 24: 281–314
32. Harjunkoski I, Jain V, Grossmann IE (2000) Hybrid mixed-integer/constrained logic programming strategies for solving scheduling and combinatorial optimization problems. Computers and Chemical Engineering 24: 337–343
33. Heipcke S (1995) Resource constrained job-shop-scheduling with constraint nets. Diploma thesis, Katholische Universität Eichstätt, Mathem.-Geographische Fakultät, Universität Eichstätt, Eichstätt
34. Heipcke S (1999) Combined modelling and problem solving in mathematical programming and constraint programming. PhD thesis, School of Business, University of Buckingham, Buckingham
35. Holland JH (1975) Adaption in natural and artificial systems. University of Michigan Press, Ann Arbor, MI
36. Holland JH (1992) Adaption in Natural and Artificial Systems. MIT Press, Cambridge, MA
37. Jain V, Grossmann IE (2001) Algorithms for hybrid MILP/CP models for a class of optimization problems. IFORMS Journal on Computing 13: 258–276
38. Kallrath J (1999) The concept of contiguity in models based on time-indexed formulations. In: Keil F, Mackens W, Voss H, Werther J (eds) Scientific computing in chemical engineering ii, pp 330–337. Springer, Berlin Heidelberg New York
39. Kallrath J (2000) Mixed integer optimization in the chemical process industry: Experience, potential and future perspectives. Chemical Engineering Research and Design 78(6): 809–822
40. Kallrath J (2002) Combined strategic and operational planning – a MILP success story in chemical industry. OR Spectrum 24: 315–341
41. Kallrath J, Schreieck A (1995) Discrete optimization and real world problems. In: Hertzberger B, Serazzi G (eds) High-performance computing and networking, pp 351–359. Springer, Berlin Heidelberg New York
42. Kallrath J, Wilson JM (1997) Business optimisation using mathematical programming. Macmillan, Houndmills, Basingstoke
43. Kocher M (2000) A greedy approach for batch processing problems from process industry. Diploma thesis, TU Graz, Graz
44. Kondili E, Pantelides CC, Sargent RWH (1988) A general algorithm for scheduling of batch operations. In: Proceedings 3rd International Symposium on Process Systems Engeneering, pp 62–75. CACHE Corp., Sydney
45. Kondili E, Pantelides CC, Sargent RWH (1993) A general algorithm for short-term scheduling of batch operations – I. MILP formulation. Computers Chemical Engineering 17: 211–227
46. Marriot K, Hallo PJ (1998) Programming with constraints – an introduction. MIT Press, Boston
47. Mendez CA, Cerda J (2002) An MILP-based approach to the short-term scheduling of make-and-pack continuous production plants. OR Spectrum 24, No. 4 (to appear)
48. Michaelewicz Z, Fogel DB (2000) How to solve it: modern heuristics. Springer, Berlin Heidelberg New York

49. Möhring RH, Schulz AS, Stork F, Uetz M (2000) Solving project scheduling problems by minimum cut computations. Technical Report 680, Fachbereich Mathematik, Technische Universität Berlin, Berlin
50. Möhring RH, Schulz AS, Stork F, Uetz M (1999) On project scheduling with irregular starting time costs. Operations Research Letters (in press); also appeared in: Nešetřil J (ed) Algorithms — ESA'99, 7th Annual European Symposium, pp 139–150. Lecture Notes in Computer Science, Vol 1643. Springer, Berlin Heidelberg New York
51. Neumann K (1975) Operations Research Verfahren, Band III. Carl Hanser, München
52. Neumann K, Schwindt C, Trautmann N (2002) Advanced production scheduling for batch plants in process industries. OR Spectrum 24: 251–279
53. Neumann K, Schwindt C, Zimmermann J (2002) Project scheduling with time windows and scarce resources. Springer, Berlin Heidelberg New York
54. Neumann K, Trautmann N, Schwindt C (2001) Short-term planning of batch plants in process industries. In: Kischka P, Leopold-Wildburger I, Möhring RH, Radermacher FJ (eds) Models, methods and decision support for management, pp 211–226. Physica, Heidelberg
55. Pantelides CC (1994) Unified frameworks for optimal process planning and scheduling. In: Proceedings 2nd Conference Foundations of Comp Aided Proc Op, pp 253–274. CACHE Corp, Sydney
56. Patterson JH, Slowinski R, Talbot FB, Weglarz J (1989) An algorithm for a general class of precedence and resource constrained scheduling problems. In: Slowinski R, Weglarz J (eds) Advances in project scheduling, pp 1–26. Elsevier, Amsterdam
57. Pritsker AAB, Watters LJ, Wolfe PM (1969) Multi project scheduling with limited resources: A zero-one programming approach. Management Science 16: 93–108
58. Rosenau B (1996) Bearbeitung eines Losgrössenproblems aus der chemischen Industrie mit Schnittebenenverfahren. Diploma thesis, University of Marburg, Marburg
59. Sand G, Engell S, Märkert A, Schultz R, Schulz C (2000) Approximation of an ideal online scheduler for a multiproduct batch plant. Computers and Chemical Engineering 24: 361–367
60. Sawik T (1987) Multilevel scheduling of multistage production with limited in-process inventory. Journal of Operational Research Society 38: 651–664
61. Schultz R (1995) On structure and stability in stochastic programs with random technology matrix and complete integer recourse. Mathematical Programming 70: 73–89
62. Schwindt C (1998) Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen Zeitfenstern. Shaker, Aachen
63. Schwindt C, Trautmann N (2000) Batch scheduling in process industries: an application of resource-constrained project scheduling. OR Spektrum 22(4): 501–524
64. Sgall J (1998) On-line scheduling. In: Fiat A, Woeginger GJ (eds) Online algorithms, pp 196–231. Springer, Berlin Heidelberg New York
65. Shah N (1999) Single- and multisite planning and scheduling: Current status and future challenges. AIChE Symposium Series 320(94): 75–90
66. Shah N, Pantelides CC, Sargent RWH (1993) A general algorithm for short-term scheduling of batch operations – 2. Computational issues. Computers Chemical Engineering 17: 229–244
67. Shah N, Pantelides CC, Sargent RWH (1993) Optimal periodic scheduling of multipurpose batch plants. Annals of Operations Research 42: 193–228
68. Timpe C (2002) Solving planning and scheduling problems with combined integer and constraint programming. OR Spectrum 24, No. 4 (to appear)
69. Trautmann N (2001) Anlagenbelegungsplanung in der Prozessindustrie. Gabler, Wiesbaden
70. Wolsey LA (1998) Integer programming. Wiley, New York