Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

SIMULATION-BASED OPTIMIZATION IN MAKE-TO-ORDER PRODUCTION: SCHEDULING FOR A SPECIAL-PURPOSE GLASS MANUFACTURER

Carsten Ehrenberg
Jürgen Zimmermann

Clausthal University of Technology
Julius-Albert-Straße 2
D-38678 Clausthal-Zellerfeld, GERMANY

ABSTRACT

We consider the problem of determining schedules for make-to-order production of companies that manufacture special purpose glasses. Due to sensitive raw materials and high quality specifications, scheduling is affected by disturbances arising from stochastic processing times and stochastic scrap rates. Scarce machine capacities, limited availability of transportation equipment, and technical or organizational temporal constraints lead to a complex planning problem. Hence, discrete-event simulation is valuable for analyzing the impact and robustness of alternative schedules, but it fails in efficiently guiding the search for optimal control parameters. In order to overcome this drawback, we propose a simulation-based optimization approach that relies on coupling simulation and optimization through a relaxation-based schedule generation procedure. Schedules are generated employing a mixed-integer programming model for which input parameters and additional constraints are iteratively derived using a simulation model. We evaluate our approach considering real-world instances and present preliminary computational results indicating its effectiveness.

1 INTRODUCTION

Since most manufacturing problems are characterized by a high level of complexity, the use of simulation models for supporting operational scheduling tasks has become more and more important in practice. In this context discrete-event simulation (DES) represents an effective tool for analyzing and evaluating impacts of alternative schedules for real world production systems. However, if the search for control parameters that leads to good system performance measures is not guided in an appropriate way, creating detailed schedules for the shop floor can require a huge number of time consuming simulation experiments. In special purpose glass manufacturing scarce resources, time-varying bottlenecks, as well as temporal constraints cause complex production systems and feasible schedules with a good performance are difficult to generate. Hence, large improvements in both factory performance and planning speed can be achieved by coupling simulation and optimization through a simulation-based optimization approach. In this paper we will present a novel approach where schedules are generated by a mixed-integer programming model for which input parameters and additional constraints are iteratively derived using a simulation model. The remainder of this paper is organized as follows. In Section 2 we introduce the considered planning problem and emphasize special characteristics of the underlying glass manufacturing process. An overview of related work is given in Section 3. In Section 4 we show how the scheduling problem can be formulated as a resource-constrained project scheduling problem. The proposed simulation-based optimization approach is presented in Section 5. We give computational results for real-world instances in Section 6 and present a reactive strategy for rescheduling in Section 7. Finally, Section 8 contains our conclusions and outlines possibilities for future research.

2 PROBLEM DESCRIPTION

The planning problem considered in this paper originates from short-term make-to-order production planning of a medium-sized special purpose glass manufacturer. The raw material glass is highly sensitive and production is especially error-prone compared to production of materials like metal or plastics. Already minor changes in partially not controllable process parameters, e. g., air-pressure, or variations in raw material properties lead to stochastic scrap-rates and considerable variability of processing times which can hardly be anticipated.

Manufacturing steps are carried out using multi-purpose machines for cutting, edge grinding, printing, laminating and tempering. Since some jobs may have to visit certain machines more than once, complex material flows with job recirculation occur. In order to provide flexibility the type of factory layout is basically a job shop with identical parallel machines. As both the product portfolio and the production volume vary frequently over time, bottlenecks change and are difficult to predict. Machines are loaded with workpieces by workers, who also operate the machines and perform quality control operations. If dimensions of glass plates exceed certain limits, at least two workers are required to handle workpieces for safety reasons. Hence, workers and their allocation to machines are important in order to run the production smoothly and their shift plans and breaks have to be considered when generating production schedules.

Because large-scale products (e. g., LCD display front panels) account for a significant percentage of the production volume, the limited availability of special transportation equipment (e. g., rack trolleys, intainers, or boxes) plays an important role in the schedule generation process. We carried out preliminary simulation studies, which show that schedules generated by the traditionally used production planning and control system often lead to production situations where operations have to be delayed by a significant amount of time as a result of missing transportation equipment. As a consequence extensive manual rescheduling activities on shop floor level have to be carried out, and cycle time increases. Further, processing of large-scale products also leads to increased space requirements caused by temporary material storage in machine input and output buffers that have to be considered in planning too.

Finally, specific temporal constraints between processing operations have to be observed. Organizational temporal constraints may, for example, originate from customer deadlines for the delivery of products or ready times for raw material availability. Technical temporal constraints are, e. g., required to guarantee production-related drying times or must be established for operations that have to be carried out one after another without any delay in between. In glass manufacturing transportation lot sizes are generally magnitudes smaller than production lot sizes. Therefore, allowing overlapping of operations leads to further temporal constraints. Overlapping means that some units of a production lot may be transferred to a successive machine where processing is started before the processing of the entire production lot on the predecessor machine is completed. Overlapping of operations can reduce the cycle time of jobs substantially, but makes scheduling even more complex because additional time lags have to be considered that affect the time windows for operations start times.

In this paper we focus on minimizing the makespan, i. e., the maximum completion time of any operation, which is a commonly used optimization goal in the domain of make-to-order production. The minimization of the makespan generally leads to improved machine utilization and a reduction in cycle times (Pinedo 2012). In conclusion, the described planning problem can be interpreted as a generalization of the job shop scheduling problem. Because of the characteristics of the glass production process (stochastic process parameters, different resource types, shift plans, transport equipment, material storage, temporal constraints) it seems appropriate to consider the problem as extended resource-constrained project scheduling problem with generalized precedence relations (RCPSP/max) under uncertainty, which is in turn a generalization of the job shop problem and belongs to the class of NP-hard problems (Neumann, Schwindt, and Zimmermann 2006). Allowing for operations to be interrupted due to shift times or breaks accounts for the first extension to the underlying problem and considering transport equipment and material storage forms the second extension.

3 RELATED WORK

Various extensions of basic scheduling problems have been developed in order to cover practical needs (Hartmann and Briskorn 2010). There is also a vast majority of literature that has primarily been focused on finding solutions for deterministic scheduling models assuming that all problem characteristics and input parameters are known (Brucker 2007, Pinedo 2012). However, this assumption is unlikely to be fulfilled in many manufacturing environments. Therefore, ongoing research aims to close the gap between scheduling theory and its applicability to industrial applications. Efforts are made to extend deterministic approaches to situations where some form of executional uncertainty (e. g., disruptions due to machine failures, arrivals of urgent jobs, changes in job processing time) can occur. A very common approach used in manufacturing systems is *predictive-reactive scheduling*, which is basically a two-step process (Ouelhadj and Petrovic 2009). First, a baseline schedule (also called predictive schedule) is developed. This baseline schedule is then modified or rescheduled during its execution in response to unexpected real-time events. A similar approach is deployed in the paper at hand. Aytug et al. (2005) and Herroelen and Leus (2005) give extensive surveys of recently developed approaches for stochastic scheduling. For a review of rescheduling techniques for manufacturing problems we refer to Vieira, Herrmann, and Lin (2003).

Simulation-based optimization is usually applied when manufacturing problems are characterized by a high level of complexity that makes it inappropriate to formulate the (entire) underlying problem analytically. In our approach we deploy a *mixed-integer programming* (MIP) model for RCPSP/max that is based on a discrete-time formulation, which can be solved by standard optimization software like CPLEX. Due to increases in computers processing power and improved algorithms it is possible to solve medium-sized real-world instances of the considered problem that way. However, if very large instances have to be tackled, appropriate heuristic scheduling procedures can easily be integrated in our approach. For a review of state-of-the-art heuristic solution procedures we refer to Kolisch and Hartmann (2006). Notice, that near-optimal solutions resulting from solving a MIP model always permit evaluating the gap between this solution and the best possible solution, whereas heuristics for the problem considered feature no performance-guarantees. In contrast to the classical RCPSP/max, variants containing the aforementioned extensions, e. g., shift plans by the concept of calendars (Franck, Neumann, and Schwindt 2001) or interruptions of operations by allowing activity splitting (Buddhakulsomsiri and Kim 2006), are virtually computational intractable and optimal or near-optimal solutions can be computed for very small instances only. Further, some characteristics of the described glass manufacturing process, especially those related to material handling activities, lead to complex interdependencies that cannot be addressed in an analytical manner, but can be integrated into a simulation model efficiently. For baseline schedule generation we therefore rely on a MIP model that contains only the fundamental structural properties of the problem considered, i. e., machines and workers as well as temporal constraints are explicitly modeled. Shift plans, activity splitting and material handling are integrated in a simulation model, which transforms the generated solutions to real-world applicable schedules. Note, that both the MIP model and the simulation model used for schedule evaluation are based on deterministic input parameters, i. e., mean values for processing times and scrap-rates. In order to account for uncertainties in processing times and scrap-rates we use the reactive strategy described in Section 7. For an overview of research on proactive-reactive scheduling, where, in contrast to predictive-reactive scheduling, statistical knowledge of uncertainty is taken into account when constructing the baseline schedule, we refer to Van de Vonder, Demeulemeester, and Herroelen (2007).

After generating and parameterizing the simulation model, an initial MIP model is build, where "difficult" constraints are omitted, i. e., the model contains only a subset of all constraints. This model consequently represents a *relaxation* of the problem described in Section 2. Schedules generated with the relaxed model are in general not feasible. Hence, with the help of consecutive simulations runs additional constraints are generated, unless the schedule can be proven to be feasible for all constraints of the considered problem instance. Related relaxation-based approaches have successfully been applied in project scheduling (Bartusch, Möhring, and Radermacher 1988), but have attracted less attention in simulation-based scheduling research yet. Byrne and Bakir (1999), Lee and Kim (2002) and Morito et al. (1999) employ similar approaches for distribution and supply chain planning.

In research literature, which is related to simulation-based scheduling for manufacturing problems basically three kinds of optimization approaches are employed (Klemmt et al. 2009): Dispatching rules, heuristic search methods, and mathematical programming. Albeit promising results concerning dispatching rules and heuristic search methods have been stated over the last years (Andersson, Ng, and Grimm 2008) we expect these approaches not to yield the performance envisaged for the considered production system. Compared to global scheduling approaches like heuristic search or mathematical programming the performance of dispatching rules is hard to predict because decisions are made locally in a myopic manner. As a result of the described temporal constraints and resource restrictions the integration of appropriate job dispatching rules and resource allocation rules for transportation equipment in a simulation model is a difficult task. In our case, several additional information items must be processed to decide which operation to schedule next, e. g., information about time lags, predicted completion times of operations, shift times or availability of equipment. We conducted preliminary simulation studies incorporating dispatching rules that revealed the problem of running into production deadlocks caused by a lack of transportation equipment which could not be anticipated based on local information. Further, since glass manufactures are confronted with a frequently changing product mix caused by decreasing life-cycles of their products, the production system is reconfigured frequently too, e. g., new machines are integrated. Hence, it can hardly be guaranteed that a type of dispatching rule, which has performed well for the old system, also performs well for the new one. Instances of the considered planning problem are characterized by a huge search space. At the same time the number of feasible solutions is quite small because of the restrictiveness resulting from scarce machines, limited transportation equipment and temporal constraints. In order to build reliable and valid simulation models of glass manufacturing processes a relatively high level of detail is required, which causes considerable execution time of simulation runs. Hence, applying randomized search strategies, where the performance of many alternative schedules is evaluated by successive simulation runs, is inapplicable due to the high computational effort. Approaches incorporating heuristic search methods to tackle scheduling problems are generally effective if a great number of rather short simulation runs can be performed to find good solutions. In contrast, to reduce computing times, our approach aims in carrying out as few simulation runs as possible until a good feasible schedule is generated.

4 OPTIMIZATION MODEL

In this section we describe the deployed optimization model. For details on network planning and project scheduling we refer to Neumann, Schwindt, and Zimmermann (2003). To present the optimization problem we use an activity-on-node network $N = (V,E;\delta)$, with node set V, arc set E and arc weights δ. We interpret the production process as project planning problem by assigning to every manufacturing operation (in what follows we use the terms "operation" and "activity" synonymously) a corresponding node $i \in V$. Node set $V := \{0,1,\ldots,n,n+1\}$ consists of n (real) activities, $1,\ldots,n$, that have to be carried out without interruption, and two fictitious activities, 0 and $n+1$, that represent the beginning and completion of the underlying project, respectively. We denote the start time of activity $i \in V$ by $S_i \in \mathbb{Z}_{\geq 0}$, its processing time by $p_i \in \mathbb{Z}_{\geq 0}$ and its completion time by $C_i := S_i + p_i$. A sequence of start times $S = (S_0,S_1,\ldots,S_{n+1})$, where $S_i \geq 0$ $(i \in V)$ and $S_0 := 0$, is termed a *schedule*. S_{n+1} equals the project duration. The complete planning period is partitioned into $\overline{d}$ time slots $[t-1,t[$ for $t = 0,\ldots,\overline{d}$. Parameter $\overline{d}$ represents an upper bound on the shortest project duration and can be defined as $\overline{d} := \sum_{i\in V} \max(p_i, \max_{\langle i,j\rangle \in E} \delta_{ij})$. We choose a hourly time pattern, i. e., the difference between two adjacent points in time $t-1$ and t $(t \in \{1,\ldots,\overline{d}\})$ is constant and amounts to one hour. For notational convenience we set $T := \{0,\ldots,\overline{d}\}$.

Between two operations $i, j \in V$, $i \neq j$ general temporal relationships can exist, that can be derived from technical or organizational needs. If, e. g., activity j cannot be started earlier than $d_{ij}^{min} \in \mathbb{Z}_{\geq 0}$ time units after activity i (minimum time lag), i. e., $S_j - S_i \geq d_{ij}^{min}$, we introduce an arc $\langle i, j\rangle$ having weight $\delta_{ij} := d_{ij}^{min}$ into network N. If activity j must be started no later than $d_{ij}^{max} \in \mathbb{Z}_{\geq 0}$ time units after activity i (maximum time lag), i. e., $S_j - S_i \leq d_{ij}^{max}$, we introduce a backward arc $\langle j,i \rangle$ with weight $\delta_{ji} := -d_{ij}^{max}$. The resulting arc set E represents the temporal constraints $S_j - S_i \geq \delta_{ij}$ among the start times of activities $i, j \in V$. The set of feasible start times of activity $i \in V$ forms a time window $\{ES_i, \ldots, LS_i\}$, where ES_i is the earliest and LS_i the latest start time of activity i with respect to the given temporal constraints.

In practice, different types of resources are required to carry out operations. In this paper we consider sets of renewable resources $\mathscr{R}$ and cumulative resources $\mathscr{R}^\gamma$, respectively. *Renewable resources* are available in each single time period of the planning horizon independently of their previous utilization. We use this kind of resource to model machinery and workers. Activity $i \in V$ requires $r_{ik} \in \mathbb{Z}_{\geq 0}$ units of resource $k \in \mathscr{R}$ during its execution. Given some schedule S, the set of activities in progress at time t, is given by $\mathscr{A}(S,t) := \{i \in V \mid S_i \leq t < C_i\}$. Thus, $r_k(S,t) := \sum_{i \in \mathscr{A}(S,t)} r_{ik}$ represents the total amount of resource $k \in \mathscr{R}$ required for those activities in progress at time $t \in T$. Each resource has a maximum capacity denoted by R_{kt}. For example, consider certain periods $\{\tilde{t}_0, \ldots, \tilde{t}_n\} \in T$ where one out of two identical parallel machines is down due to maintenance. We model this situation by setting $R_{kt} := 1$ $(k \in \mathscr{R}, t \in T \cap \{\tilde{t}_0, \ldots, \tilde{t}_n\})$ and $R_{kt} := 2$ $(k \in \mathscr{R}, t \in T \setminus \{\tilde{t}_0, \ldots, \tilde{t}_n\})$, respectively (note, that k is the proper resource index). For a schedule S to be feasible, the amount of resources required must not exceed the capacity of resources, i. e., $r_k(S,t) \leq R_{kt}$ $(k \in \mathscr{R}, t \in T)$. Transportation equipment and storage facilities are modeled by using so-called *cumulative resources*, which have given maximum quantities R_k^γ (Bartels and Zimmermann 2009, Neumann and Schwindt 2003). In contrast to renewable resources, the availability of cumulative resources at a certain point in time depends on the history of the manufacturing process as they are used and released over time. (Note, that in literature, renewable resources with $R_{kt} = 1$ are sometimes called "disjunctive resources", whereas the name "cumulative resource" is then used when referring to renewable resources with capacities $R_{kt} > 1$.) Typically, associated with the start of operation i, say, $r_{ik} > 0$ units of cumulative resource $k \in \mathscr{R}^\gamma$ are needed in order to store workpieces. Likewise, at the completion of activity i, $-r_{ik'} > 0$ units of cumulative resource $k' \in \mathscr{R}^\gamma$ are released. By $V_k^+ := \{i \in V | r_{ik} > 0\}$ and $V_k^- := \{i \in V | r_{ik} < 0\}$ we denote the disjoint sets of activities using and releasing resource $k \in \mathscr{R}^\gamma$. In the event that activity $i \in V$ both uses and releases, say, $\tilde{r}_{ik}$ units of one and the same cumulative resource $k \in \mathscr{R}^\gamma$ (e. g., a circular storage system at a printing machine), we split up activity i into two new ones, say, i' and i'' with $p_{i'} = p_{i''} := p_i$ and $r_{i'k} = -r_{i''k} := \tilde{r}_{ik}$. Further, we add arcs $\langle i', i'' \rangle$ and $\langle i'', i' \rangle$ with weight $\delta_{i'i''} = 0$ and $\delta_{i''i'} = 0$ to project network N. In doing so, for any feasible schedule, $S_{i'} = S_{i''}$ holds. Then, during time interval $[S_{i'}, C_{i''}[$, exactly $\tilde{r}_i$ units of cumulative resource k are occupied (note that $C_{i''} = S_{i'} + p_{i'}$). Given a schedule S, the *active set*, i. e., the set of activities that have used resource $k \in \mathscr{R}^\gamma$ by time $t \geq 0$, is given by $\mathscr{A}_k^\gamma(S,t) := \{i \in V_k^+ | S_i \leq t\} \cup \{i \in V_k^- | C_i \leq t\}$ $(k \in \mathscr{R}^\gamma, t \in T)$. Thus, the term $r_k^\gamma(S,t) := \sum_{i \in \mathscr{A}_k^\gamma(S,t)} r_{ik}$ represents the units of resource $k \in \mathscr{R}^\gamma$ in use at time $t \in T$. We call function $r_k^\gamma(S,\cdot)$ the *demand profile* of resource $k \in \mathscr{R}^\gamma$. Schedule S is feasible if it satisfies the constraints $r_k^\gamma(S,t) \leq R_k^\gamma$ $(k \in \mathscr{R}^\gamma, t \in T)$.

Considering a *discrete-time* formulation (Pritsker, Watters, and Wolfe 1969) with binary variables x_{it} that allocate feasible start times $t \in T$ to activities $i \in V$, i. e., $x_{it} := 1$, if activity i starts at time t and $x_{it} := 0$ otherwise, the resource-constrained project scheduling problem with cumulative resources and generalized precedence relations $PSc|temp|C_{max}$ can be formulated as a mixed-integer programming model as follows:

$$\text{Minimize} \quad \sum_{t=ES_{n+1}}^{LS_{n+1}} t\, x_{n+1,t} \tag{1}$$

$$\text{subject to} \quad \sum_{t=ES_i}^{LS_i} x_{it} = 1 \qquad (i \in V) \tag{2}$$

$$\sum_{t=ES_j}^{LS_j} t\, x_{jt} - \sum_{t=ES_i}^{LS_i} t\, x_{it} \geq \delta_{ij} \qquad (\langle i,j \rangle \in E) \tag{3}$$

$$\sum_{i \in V} r_{ik} \sum_{\tau=\max\{ES_i,\ t-p_i+1\}}^{\min\{t,\ LS_i\}} x_{i\tau} \leq R_{kt} \qquad (k \in \mathscr{R}, t \in T) \tag{4}$$

$$\sum_{i \in V_k^+} r_{ik} \sum_{\tau=0}^{t} x_{i\tau} + \sum_{i \in V_k^-} r_{ik} \sum_{\tau=0}^{t-p_i} x_{i\tau} \leq R_k^\gamma \qquad (k \in \mathscr{R}^\gamma, t \in T) \tag{5}$$

$$x_{00} = 1 \tag{6}$$

$$x_{it} \in \{0,1\} \qquad (i \in V, t \in \{ES_i, \ldots, LS_i\}) \tag{7}$$

Objective function (1) minimizes the start time of the terminating activity, i. e., the makespan. Constraints (2) and (7) force each activity to receive exactly one start time. Since $S_i = \sum_{t \in \{ES_i,\ldots,LS_i\}} t\, x_{it}$ for $i \in V$, inequalities (3) guarantee that the temporal constraints will be satisfied. Restrictions (4) and (5), respectively, ensure that the renewable and cumulative resource constraints are satisfied. Condition (6) sets the start time for the project to zero. For alternative modeling techniques employing SAT-solvers or constraint propagation algorithms we refer to Horbach (2010) and Laborie (2003).

5 SIMULATION-BASED OPTIMIZATION APPROACH

In what follows, we are going to present the proposed simulation-based optimization approach. The framework of the approach is schematically shown in Figure 1.

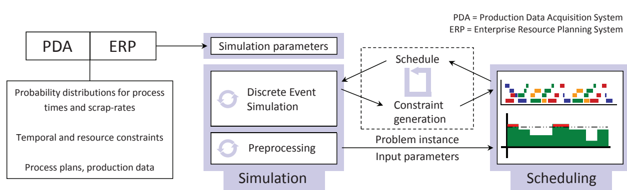


Figure 1: Simulation-based optimization framework.

A simulation model is based on several input parameters (e. g., probability distributions for processing times and scrap rates) as well as the configuration of the production system. Modeling is carried out data-driven by using a generation procedure that is implemented in the ModL programming language of the simulation software *ExtendSim*. All the data needed for a simulation run is stored in a database (internal ExtendSim database), so the simulation model is (at least conceptually) separated from the simulation software. The generation procedure builds a simulation model that represents the actual planning problem by combining predefined meta-blocks (machine-blocks, transport-blocks, and dispatching-blocks). Then, during a preprocessing-step, the meta-blocks are configured; e. g., a block representing a machine is linked to the database table that contains its shift-plan. Other configuration steps are carried out dynamically by items (e. g., jobs) at simulation run time. For this purpose every created job has an additional attribute that references an database entry point. Starting from that entry point jobs can access job-related (e. g., due-dates) and job-resource-related (e. g., processing times, resource demands, starting times) parameters. Notice, the possibility to use job-resource-related parameters to (re-)configure meta-blocks, e. g., consider a machine block that switches from batch-processing to single-item-processing *et vice versa*. Once the simulation model has been generated, the MIP model can be build. Thereby, the set of operations V is derived from information about customer orders within the planning horizon. As parameters p_i $(i \in V)$ depend on probability distributions for both processing times and scrap rates, we carry out preprocessing simulation runs in order to estimate the values used in the optimization model (i. e., the mean values). Resource requirements r_{ik} as well as capacities R_{kt} and R_k^γ $(i \in V, k \in \mathscr{R} \cup \mathscr{R}^\gamma, t \in T)$ follow from process plans and the number of available machines, workers, and transportation equipment. Time lags δ_{ij} $(\langle i,j \rangle \in E)$ result, among others, from customer due dates, delivery dates for raw materials, process plans, and consideration of overlapping between operations. Suppose, e. g., that due to a customer order, some job has to be completed by time t' at the latest and raw material needed to start this job is available at time t'' at the earliest. Let $\{i', j', \ldots, h'\} \in V$ be the operations required to perform this job. Activity h' then has to be begun a period of time $d^{max}_{0h'} := t' - p_{h'}$ after the beginning of the project at the latest. Similarly, activity i' can be begun $d^{min}_{0i'} := t''$ after the beginning of the project at the earliest. Thus, arc set E is extended by arcs $\langle h', 0\rangle$ and $\langle 0, i'\rangle$ with weights $\delta_{h'0} := -t' + p_h$ and $\delta_{0i'} := t''$, respectively, and proper temporal constraints (3) are added to the MIP model. Figure 2 illustrates how time lags for activity overlapping are derived.

Operation i	Operation j	Legend:
$q_i = 75$ items	$q_j = q_i \cdot (1 - \sigma_i) = 62$ items	q : Production lot size
$\tau_i = 10$ minutes/item	$\tau_j = 5$ minutes/item	τ : Processing time per item
$\vartheta_i = 15$ minutes	$\vartheta_j = 60$ minutes	ϑ : Machine setup-time
$\sigma_i = 17\%$	$\sigma_j = 20\%$	σ : Scrap-rate
$p_i = (q_i \tau_i) + \vartheta_i = 765$ minutes	$p_j = (q_j \tau_j) + \vartheta_j = 370$ minutes	z : Transportation lot size
$z_i = 25$ items	$z_j = 10$ items	

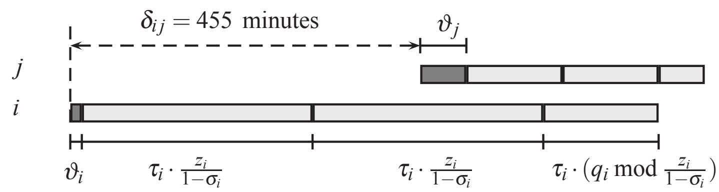


Figure 2: Overlapping operations with minimum time lag.

Operations j is slower than operation i, i. e., $p_i > p_j$. In order to start operation j once the first transport lot is available, and then process it without interruption, preparing resources required to carry out j should be started at the right time. Here, a minimum time lag between operations i and j is introduced. Analogously, time lags are constructed for $p_i \leq p_j$.

Schedule generation is carried out through an iterative process of solving a *resource-relaxation* of problem $PSc|temp|C_{max}$, and evaluating its solution with the help of the simulation model. The initial MIP model is obtained by omitting cumulative resource constraints (5), i. e., problem $PS|temp|C_{max}$ is considered. An optimal (or near-optimal) solution S' of the resource-relaxation is in general resource-infeasible related to the original planning problem. At certain points in time $t \in T$, demand for at least one resource $k \in \mathscr{R}^\gamma$ exceeds the quantity available, i. e., a so-called *resource conflict* $\sum_{i \in \mathscr{A}^\gamma(S',t)} r_{ik} > R^\gamma_k$ for some $k \in \mathscr{R}^\gamma$ occurs. This conflict is caused by the simultaneous execution of operations $i \in \mathscr{A}^\gamma(S', t)$ and can be resolved by introducing proper constraints, which prevent activities from all being in progress at the same time. Suppose schedule S' is a solution of $PS|temp|C_{max}$ and simulation shows at least one resource conflict. If more than one resource conflict exists, we identify *one* resource $k' \in \mathscr{R}^\gamma$ and the point in time t^P, where the *peak demand* occurs, i. e., $r^\gamma_{k'}(S, t^P) = \max_{k \in \mathscr{R}^\gamma, t \in T} r^\gamma_k(S, t)$. At time t^P we then determine those two operations, say, j' and j'' that account for the largest amount of the peak demand, i. e., $r_{j'k'} \geq r_{j''k'} \geq r_{hk'}$ for all operations $h \in V \setminus \{j', j''\}$. Let operations $\underline{j'}$ and $\underline{j''}$ be the two operations that use resource k' foremost and $\overline{j'}$ and $\overline{j''}$ be the two operations that finally release resource k'. Further, with j'_0 and j''_0 ($j'_{n'}$ and $j''_{n''}$) we denote the first (last) operations of the jobs $J' := \{j'_0, \ldots, \underline{j'}, \ldots, j', \ldots, \overline{j'}, \ldots, j'_{n'}\}$ and $J'' := \{j''_0, \ldots, \underline{j''}, \ldots, j'', \ldots, \overline{j''}, \ldots, j''_{n''}\}$, i. e., the jobs containing operations j' and j''. In order to achieve a feasible solution, a *conflict set* $F := J'_{k'} \cup J''_{k'}$ with $J'_{k'} := \{\underline{j'}, \ldots, \overline{j'}\} \subseteq J'$ and $J''_{k'} := \{\underline{j''}, \ldots, \overline{j''}\} \subseteq J''$ has to be broken up. Usually this is done by considering *disjunctive precedence relations* (Pinedo 2012). In the present case, the time lag between $S_{\underline{j''}}$ and $S_{\overline{j'}}$ ($S_{\underline{j'}}$ and $S_{\overline{j''}}$), which is needed in order to establish a precedence relation, is not only affected by operations processing times, but also by waiting times that depend on the realized schedule, and is therefore unknown in advance. Hence, because introducing precedence relations to break up resource conflicts is not possible, we apply the following special type of resource constraint:

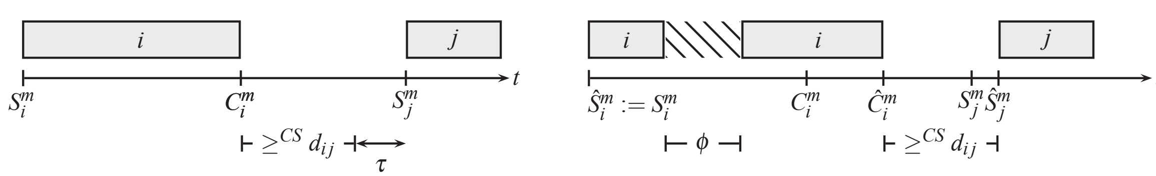
$$\sum_{i \in F} r_{ik} \leq 1 \quad (k \in \mathscr{R}^O) \tag{8}$$

Set $\mathscr{R}^O$ contains "dummy resources" for ordering purposes, which can be interpreted as a special kind of cumulative resource with capacity $R_k^O := 1$ $(k \in \mathscr{R}^O)$. Let $\tilde{k} \in \mathscr{R}^O$ be the index according to conflict set F. By choosing appropriate resource requirements, i. e., $r_{i\tilde{k}} := 1$ for $i \in \{\underline{j'}, \underline{j''}\}$, $r_{i\tilde{k}} := -1$ for $i \in \{\overline{j'}, \overline{j''}\}$ and $r_{i\tilde{k}} := 0$ for $i \in F \setminus \{\underline{j'}, \underline{j''}, \overline{j'}, \overline{j''}\}$, we force operations in F to be carried out consecutively. Integrating "order restrictions" (8) in the MIP model using equations (5) is straightforward. Summarizing, we give an algorithmic description of the proposed simulation-based optimization algorithm.

Algorithm SSG (Simulation-based schedule generation)

```
m := 0, F^m := {0} (* Initialization *)
Create simulation model; Determine N = (V,E;δ), p_i, r_ik and R_kt (i ∈ V, k ∈ ℛ, t ∈ T)
Initialize constraint set CS^m := {(2) ∪ (3) ∪ (4) ∪ (6) ∪ (7)} and MIP model MIP^m := CS^m ∪ (1)
while F^m ≠ ∅ do
    S^m ← Solve(MIP^m)
    Ŝ^m, r_k^γ(Ŝ^m, ·) ← Simulate(S^m) (* Determine demand profiles for cumulative resources k ∈ ℛ^γ *)
    if r_k^γ(Ŝ^m, t) > R_k^γ for at least one resource k ∈ ℛ^γ and one point in time t ∈ T then
        Determine t^P, k', r_ik' (i ∈ V, k' ∈ ℛ^γ) and operations j' and j''
        F^m := J'_k' ∪ J''_k' ← ConflictSet(j', j'')
        CS^m ← OrderRestrictions(J'_k', J''_k') (* Create dummy resource k^m ∈ ℛ^O and constraints (8) *)
        m := m + 1; MIP^m := MIP^(m-1) ∪ CS^m
    else
        F^m := ∅ (* Ŝ^m is a feasible solution *)
return Ŝ* := Ŝ^m
```

A call to **Simulate**(S^m) maps schedule S^m that was obtained by solving model MIP^m to the real world production system, i. e., additional constraints and properties from the underlying production system are considered. Functions **ConflictSet**() and **OrderRestrictions**() are invoked at the interface between the simulation model and the optimization model. These functions create the MIP model to be solved in the next iteration and post the conflict set information to the simulation model. To translate the results from **Solve**(MIP^m) to the simulation model, we use both the job starting times S_i^m $(i \in V)$ and the precedence relations that can be derived from S^m. Precedence relations in activity set V (i. e., $S_j \geq C_i$ for jobs $i, j \in V$) are stored in an internal database and can be accessed by dispatching modules that avoid jobs from a conflict set to be executed simultaneously. Further, starting times S_i^m serve as earliest possible starting times for the simulation of jobs. Notice, that in order to guarantee feasibility of a simulated schedule, simulation starting times $\hat{S}_i^m$ can differ from optimization starting times S_i^m (see Figure 3). Suppose, for example, that due to technical requirements (e. g., drying times) a minimum *completion-to-start time lag* ${}^{CS}d_{ij}$ between jobs $i, j \in V$ is prescribed, i. e., $S_j \geq C_i + {}^{CS}d_{ij}$.



(a) Schedule S^m from **Solve**(MIP^m). (b) Adjustment of simulation starting times.

Figure 3: Transfer of optimization results to the simulation model.

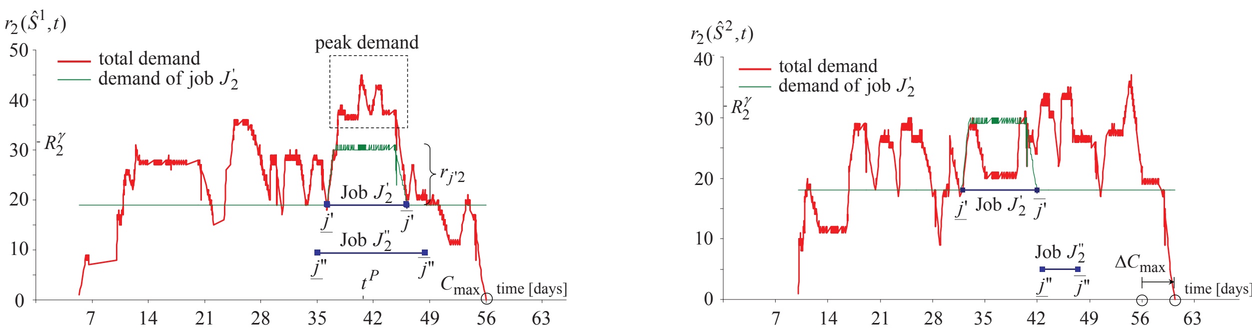
Further, the simulation model contains an additional constraint that ensures machine tool calibration after a certain number of parts have been processed. As a result job i has to be interrupted for ϕ time units. Hence, in order to maintain feasibility, the starting time of job j has to be adjusted, i. e., $\hat{S}_j^m = S_j^m + \phi - \tau$. Notice, that the temporal constraint is not binding for S^m, so the *slack* τ has to be considered when adjusting the starting time of job j.

6 PRACTICAL APPLICATION AND COMPUTATIONAL RESULTS

In this section we illustrate our approach by means of a real-world problem instance and give some preliminary computational results on its application. The algorithm for schedule generation was implemented using the simulation software *ExtendSim 8.0.2* and *CPLEX 12.3* as solver. Computations were performed on an Intel i7-980X computer with 12 GB RAM running on Windows 7 64-bit as operating system. The problem instance considered in the following consists of manufacturing four different kinds of large display front panels (ranging from 32” to 55”) and a total of 4,100 items to be processed. 140 operations and three types of transport equipment as well as 20 machines are required. All operations have to be carried out within a planning horizon of 65 days. Table 1 lists run times needed in order to achieve a feasible schedule. Notice that in this problem instance only transport equipment with index $k = 2$ and $R_2^\gamma = 32$ is restrictive. A feasible solution could be generated in less than 20 minutes, indicating the practical applicability of the proposed approach. Figure 4 shows in more detail the resolution of the conflict set occurring in iteration $m = 1$. Conflict set $\{J_2', J_2''\}$ is broken up by ensuring operations of J_2' to precede operations of J_2''.

Table 1: Computational results for the real-world problem instance.

iteration m	C_{max} [h]	run times [sec]		peak demand
		Solve(MIP^m)	**Simulate**(S^m)	$\max_{t \in T} r_2(\hat{S}^m, t)$
0	1329	160	109	67
1	1344	165	114	45
2	1419	199	112	38
3	1447	159	110	32



(a) Conflict set $\{J_2', J_2''\}$ in iteration $m = 1$.

(b) Resolution of conflict set.

Figure 4: Resource conflict.

We conducted preliminary performance studies, where we chose problem parameters in order to generate instances, which cover a wide range of practical application. The proposed approach was tested for different product mixes and production volumes leading to varying numbers of operations and resources, i. e., $|V| = \{50, \ldots, 200\}$, $|\mathscr{R}| = \{10, \ldots, 45\}$ and $|\mathscr{R}^\gamma| = \{1, \ldots, 5\}$. Further, we considered different lengths of the planning horizon and instances with rather strict or loose customer deadlines. It has turned out that, due to the used discrete-time formulation, parameters $|V|$ and $\overline{d}$ have a strong impact on the solvability of a problem instance. In fact, already the model setup and presolve process, which is carried out by the solver, requires significant time and memory for larger instances (i. e., $\overline{d} \geq 10$ weeks, $|V| \geq 175$).

In contrast, problem hardness is less affected by parameter $|\mathscr{R}|$. This can be explained considering the so-called *resource factor RF* that reflects the average fraction of the number of resources used per activity (Kolisch, Sprecher, and Drexl 1995). If $RF = 1$ then each operation requests all resources. $RF = 0$ indicates that no job requests any resource. Computation time for solving $PS|temp|C_{max}$ increases as RF increases. In our case RF generally takes relative small values ($RF \in [0.05, 0.15]$).

In the event that operations time windows are expanded due to broad customer deadlines, the number of iterations required to find feasible solutions increases. This results from the way of creating order restrictions. Consider conflict sets F^m, F^{m+1}, F^{m+2} occurring in consecutive iterations. If time windows are tight, usually $F^m \cap F^{m+2} = \emptyset$ holds. In contrast, if time windows are large, the conflict sets are likely to have common elements, i. e., $F^m \cap F^{m+2} \neq \emptyset$. In other words, breaking up conflict sets F^m and F^{m+1} can lead to a new conflict set F^{m+2} that consists of operations from F^m and F^{m+1}. Notice, that setting $|\mathscr{R}^O| = 1$ in equations (8), always leads to disjunctive conflict sets in iterations m and $m+2$, but increases computation times required to solve the corresponding MIP models (because of the increasing resource factor of resource $k \in \mathscr{R}^O$). Clearly, the more cumulative resources are considered the more iterations are required. However, because breaking up a certain conflict set F for resource $k' \in \mathscr{R}^\gamma$ can avoid resource conflicts of the remaining resources $k \in \mathscr{R}^\gamma \setminus k'$, iterations do not increase linearly with $|\mathscr{R}^\gamma|$. This is due to temporal time lags between operations $j \in F$ and the remaining operations, which cause operations $i \in V \setminus F$ to be shifted in time too, when F is resolved.

7 REACTIVE PLANNING

Recall, that the glass manufacturing process is affected by stochastic events, e. g., stochastic scrap-rates and stochastic processing times, which can hardly be anticipated. As a consequence a schedule, which is based on estimated data (e. g., mean values for processing times and scrap-rates) may become infeasible when released to the shop floor. To overcome this problem, we use a predictive-reactive approach. First, a (predictive) baseline schedule $\hat{S}^*$ is generated with the help of algorithm **SSG**. To maintain *resource feasibility* when executing $\hat{S}^*$, the operation sequences $\mu_k := \{i_k^0, i_k^1, \ldots, i_k^{n_k}\}$ $(i \in V_k, k \in \mathscr{R} \cup \mathscr{R}^\gamma)$, which are implicitly given by $\hat{S}^*$, where $V_k := \{i \in V | r_{ik} > 0\}$ and $\hat{S}^*_{i_k^h} \leq \hat{S}^*_{i_k^{h+1}}$ for $h = 0, \ldots, n_k - 1$, are considered to be fixed and will not be changed when rescheduling is carried out. In contrast, as *time feasibility* depends on the realization of processing times and scrap rates, starting times have to be rescheduled during execution. Assume the realized value of the scrap-rate of operation i to be higher than the planned value (compare Figures 2 and 5). Thus, starting operation j at $\hat{S}^*_j$ with respect to δ_{ij}, as originally planned, will lead to starving of operation j, undesirable idle-times and additional setup-costs.

Operation i
$q_i = 75$ items
$\tau_i = 10$ minutes/item
$\vartheta_i = 15$ minutes
$\sigma_i =$ **37.5 %** (plan = 17%)
$p_i = (q_i \tau_i) + \vartheta_i = 765$ minutes
$z_i = 25$ items

Operation j
$q_j = q_i \cdot (1 - \sigma_i) =$ **47 items**
$\tau_j = 5$ minutes/item
$\vartheta_j = 60$ minutes
$\sigma_j = 20\%$
$p_j = (q_j \tau_j) + \vartheta_j =$ **295 minutes**
$z_j = 10$ items

Legend:
q : Production lot size
τ : Processing time per item
ϑ : Machine setup-time
σ : Scrap-rate
z : Transportation lot size

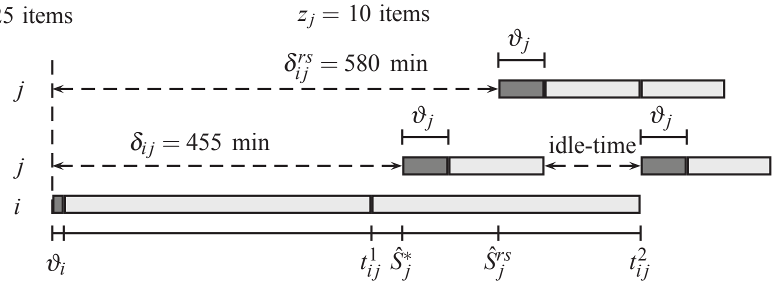


Figure 5: Rescheduling of operations start times.

In light of the practical application, we restrict the periods in time, where reactive rescheduling techniques can be applied. Rescheduling actions can only take place, when a transportation lot, that was completed by operation i is ready for operation j. The *rescheduling points* are denoted by $t^1_{ij}, \ldots, t^{nl}_{ij}$, where nl is the number of transport lots transported from i to j. Assuming that scrap items are evenly distributed within a production lot and that the mean of the realized processing times of items already processed is appropriate to estimate processing times of the remaining items, new time lags δ^{rs}_{ij} can be computed, which lead to new start times $\hat{S}^{rs}_j$.

8 CONCLUSIONS AND FUTURE RESEARCH

We presented a simulation-based scheduling approach, which relies on iteratively solving resource-relaxations of the underlying planning problem. Preliminary computational studies indicated both the effectiveness of the approach and its applicability to real-world planning situations. Further, we showed how to extend the approach by rescheduling techniques in order to cope with stochastic effects. Some of the described characteristics of glass manufacturing (e. g., temporal constraints, transportation equipment), can also be identified in other production systems. Thus adopting our approach to a broader field of application seems worthwhile. An important area of future research is the generation of *robust schedules* and the refinement of rescheduling techniques applied. A further challenging issue is to reduce the times needed to generate solutions. Here, besides the use of heuristics from the field of project scheduling, so-called *fix-and-optimize* techniques can be applied, where decisions made on former steps are fixed on successive steps. Moreover, improved methods for integrating temporal constraints in simulation models should be investigated.

ACKNOWLEDGMENTS

This research has been supported by the *Europäischer Fonds für regionale Entwicklung* (EFRE) under Grant W2-80115101.

REFERENCES

Andersson, M., A. Ng, and H. Grimm. 2008, December. “Simulation optimization for industrial scheduling using hybrid genetic representation”. In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Moench, O. Rose, T. Jefferson, and J. W. Fowler, 2004–2011. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Aytug, H., M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. 2005. “Executing production schedules in the face of uncertainties: A review and some future directions”. *European Journal of Operational Research* 161 (1): 86–110.

Bartels, J.-H., and J. Zimmermann. 2009. “Scheduling tests in automotive R&D projects”. *European Journal of Operational Research* 193 (3): 805–819.

Bartusch, M., R. H. Möhring, and F. J. Radermacher. 1988. “Scheduling project networks with resource constraints and time windows”. *Annals of Operations Research* 16 (1): 201–240.

Brucker, P. 2007. *Scheduling Algorithms*. 5th ed. Berlin, Heidelberg: Springer.

Buddhakulsomsiri, J., and D. S. Kim. 2006. “Properties of multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting”. *European Journal of Operational Research* 175 (1): 279–295.

Byrne, M., and M. Bakir. 1999. “Production planning using a hybrid simulation-analytical approach”. *International Journal of Production Economics* 59 (1–3): 305–311.

Franck, B., K. Neumann, and C. Schwindt. 2001. “Project scheduling with calendars”. *OR Spectrum* 23 (3): 325–334.

Hartmann, S., and D. Briskorn. 2010. “A survey of variants and extensions of the resource-constrained project scheduling problem”. *European Journal of Operational Research* 207 (1): 1–14.

Herroelen, W., and R. Leus. 2005. "Project scheduling under uncertainty: Survey and research potentials". *European Journal of Operational Research* 165 (2): 289–306.

Horbach, A. 2010. "A Boolean satisfiability approach to the resource-constrained project scheduling problem". *Annals of Operations Research* 181 (1): 89–107.

Klemmt, A., S. Horn, G. Weigert, and K.-J. Wolter. 2009. "Simulation-based optimization vs. mathematical programming: A hybrid approach for optimizing scheduling problems". *Robotics and Computer-Integrated Manufacturing* 25 (6): 917–925.

Kolisch, R., and S. Hartmann. 2006. "Experimental investigation of heuristics for resource-constrained project scheduling: An update". *European Journal of Operational Research* 174 (1): 23–37.

Kolisch, R., A. Sprecher, and A. Drexl. 1995. "Characterization and generation of a general class of resource-constrained project scheduling problems". *Management Science* 41 (10): 1693–1703.

Laborie, P. 2003. "Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results". *Artificial Intelligence* 143 (2): 151–188.

Lee, Y. H., and S. H. Kim. 2002. "Production-distribution planning in supply chain considering capacity constraints". *Computers & Industrial Engineering* 43 (1–2): 169–190.

Morito, S., J. Koida, T. Iwama, M. Sato, and Y. Tamura. 1999, December. "Simulation-based constraint generation with applications to optimization of logistic system design". In *Proceedings of the 1999 Winter Simulation Conference*, edited by P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. Evans, 531–536. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Neumann, K., and C. Schwindt. 2003. "Project scheduling with inventory constraints". *Mathematical Methods of Operations Research* 56 (3): 513–533.

Neumann, K., C. Schwindt, and J. Zimmermann. 2003. *Project Scheduling with Time Windows and Scarce Resources*. 2nd ed. Berlin, Heidelberg, New York: Springer.

Neumann, K., C. Schwindt, and J. Zimmermann. 2006. "Resource-Constrained Project Scheduling with Time Windows". In *Perspectives in Modern Project Scheduling*, edited by J. Józefowska and J. Weglarz, Volume 92 of *International Series in Operations Research & Management Science*, 375–407. New York: Springer.

Ouelhadj, D., and S. Petrovic. 2009. "A survey of dynamic scheduling in manufacturing systems". *Journal of Scheduling* 12 (4): 417–431.

Pinedo, M. L. 2012. *Scheduling: Theory, Algorithms, and Systems*. 4th ed. Boston, MA: Springer New York.

Pritsker, A. A. B., L. J. Watters, and P. M. Wolfe. 1969. "Multi-project scheduling with limited resources: A zero-one programming approach". *Management Science* 16 (1): 93–108.

Van de Vonder, S., E. Demeulemeester, and W. Herroelen. 2007. "A classification of predictive-reactive project scheduling procedures". *Journal of Scheduling* 10 (3): 195–207.

Vieira, G. E., J. W. Herrmann, and E. Lin. 2003. "Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods". *Journal of Scheduling* 6 (1): 39–62.

AUTHOR BIOGRAPHIES

CARSTEN EHRENBERG studied industrial engineering and business administration at the Clausthal University of Technology. Since 2009 he is research assistant at the Operations Research Group, Clausthal University of Technology. His research interests include simulation and optimization of manufacturing processes and project scheduling. His e-mail address is carsten.ehrenberg@tu-clausthal.de.

JÜRGEN ZIMMERMANN received his Diploma in Applied Mathematics and his PhD at the University of Karlsruhe, where he also completed his Habilitation. He is professor of Operations Research at the Clausthal University of Technology since 2002. His research interests include project management and scheduling, machine scheduling, and routing problems. His e-mail address is juergen.zimmermann@tu-clausthal.de.