*Simulation in Produktion und Logistik*
*Entscheidungsunterstützung von der Planung bis zur Steuerung*
Wilhelm Dangelmaier, Christoph Laroque & Alexander Klaas (Hrsg.)
Paderborn, HNI-Verlagsschriftenreihe 2013

# Simulation-based Optimization for Complex Assembly Lines with Workforce Constraints

## *Simulationsbasierte Optimierung für komplexe Montagesysteme zur personalorientierten Ablaufplanung*

Evangelos Angelidis, Daniel Bohn, Oliver Rose, Universität der Bundeswehr München, Neubiberg (Germany), evangelos.angelidis@unibw.de, daniel.bohn@unibw.de, oliver.rose@unibw.de

Sebastian Carl, TU Dresden, Dresden (Germany), sebastian.carl@tu-dresden.de

**Abstract:** The rising product diversity as well as the constantly increasing complexity of the products and assembly processes cause immense challenges to efficient planning and scheduling. The manual assembly of products with high diversity in one-of-a-kind or small batch productions by the use of multi-skilled resources can be stated as a Multi-Mode Resource-constrained Multi-Project Scheduling Problem (MMRCMPSP) with activity splitting. It is a combinatorial order problem, which is NP-hard and therefore not efficiently solvable within short runtimes for realistically problem sizes. Over the last decade, a variety of international publications indicate that there is an intensive search for solution approaches. In this paper, the authors present a simulation-based optimization platform. We implemented a variety of heuristic algorithms that generate near-optimal solutions that fulfil different predefined objective functions and constraints. We present a heuristic with emphasis on minimized slack, balanced workforce with minimal amount of elements and concurrently highest resource utilization.

## 1 Introduction

Modern assembly processes are characterised by complexity, product diversity, varying and alternative use of personnel and constantly changing conditions. This causes immense challenges to the efficient operational planning and scheduling. Existing potentials of flexibility and productivity especially in the use of resources are still not exploitable. Therefore we developed a simulation-based optimization platform (SBOP). Our main objective is the enhancement of the efficiency of manual assembly of products with high diversity for one-of-a-kind or small batch production, which is able to deal with the considerable dynamics in the use of personnel – especially if multi-skilled resources are required. By calculating optimal allocation variants, the best assembly plan for current objectives is generated for possible production scenarios, even in the case of resource outages. As a result, there will be a significant increase in productivity since the assembly cycle time as well as the manpower requirements decreases.

Complex assembly processes usually have a strongly networked process structure with a large number of varying activities, competing for resources and subject to precedence relations. Moreover flexibly applicable manual and technical resources of different skill-levels are demanded by associated production systems (Evers 2002).

The relationship between the used amount of personnel resources and the resulting duration can be represented by a capacity characteristic line for each activity (*Figure 1*). Due to technological restrictions there is a specific range between the minimum of the necessary and the maximum of the applicable resources for each assembly activity. This variation of personnel allocations leads to multiple execution modes for specific assembly activities which allow a cycle time alignment for the entire system.

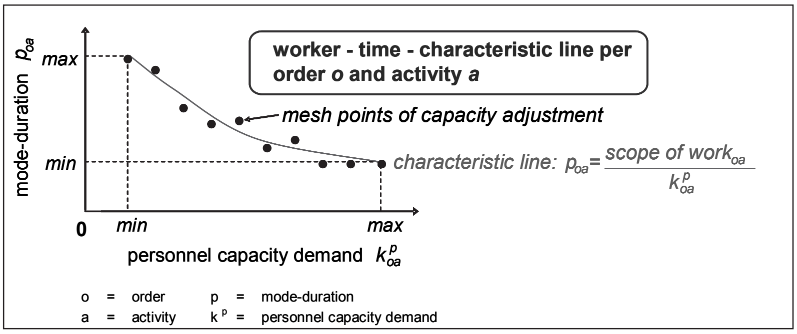


***Figure 1:*** *Example of a capacity characteristic line (Majohr 2008)*

A production scenario is a certain planning horizon with different orders (projects). They do have associated earliest start and latest finishing dates and consist of activities, subject to precedence relations. All orders as well as activities compete for limited, multi-skilled resources. This type of problem is known as the multi-mode resource-constrained multi-project scheduling problem (MRCMPSP). A wide variety of optimization procedures and solution approaches for the project scheduling have been suggested over the last years. Apart from special applications only a few of the available exact and heuristic procedures in literature achieved relevancy to practice. Many specifics of real production fields are often excluded.

The objective of this paper is to introduce our developed simulation-based optimization platform (SBOP). The SBOP is suitable for different kinds of assembly processes and offers a variety of heuristic algorithms and priority rules. Therefor we are able to generate near-optimal production plans that fulfil different predefined objective functions (minimized cost, minimized slack, balanced workforce, maximized resource utilization) and constraints.

The paper is organized as follows: In Section two we introduce the Simulation-based Optimization Platform (SBOP). Section three is devoted to our implementation of a simulation tool for complex assembly lines. Section four provides an overview of a new manufacture cost function. A heuristic algorithm is presented in Section five. Section six contains the results and a discussion of future research.

## 2 Simulation-based Optimization Platform

In 2010 a framework for Simulation-based Optimization of assembly lines was presented in (Pappert et al. 2010). That framework aimed to offer the possibility to create various solution strategies for problems in this area while reducing complexity and development time. This concept was adjusted by making broader applications and as a result created a new tool called the Simulation-based Optimization Platform (Angelidis et al. 2012a; Angelidis et al. 2012b). The new platform is based on OSGi as a runtime environment. OSGi was selected because it offers many specifications, associated implementations and it is state of the art in software development. The platform components can communicate per event (sync or out of sync) or via direct service call between each other. The existing remote interface specification is also crucial to this research due to the fact that it allows the extension of this platform with remote interface in short time and with minimal effort. The platform has a flexible and easy modifiable meta-data model for working with several companies of different size and assembly approaches. Web technology and cloud services provide a user-friendly interface for uploading factory plans, triggering optimization runs with different parameters and visualizing the optimization results. The optimization approach in SBOP is a cycle (illustration in *Figure 2*) of the following steps:

- Creation of a scenario based on initial setups or results of previous optimization steps.
- Simulation of this scenario.
- Data preparation of the simulation results (Key Performance Indicators).
- Evaluation of simulation results.

This cycle is terminated when the target criteria or an abort criteria is reached. As a central element within SBOP, the optimizer module is responsible for the creation of scenarios which we want to simulate and the evaluation of the simulation results.

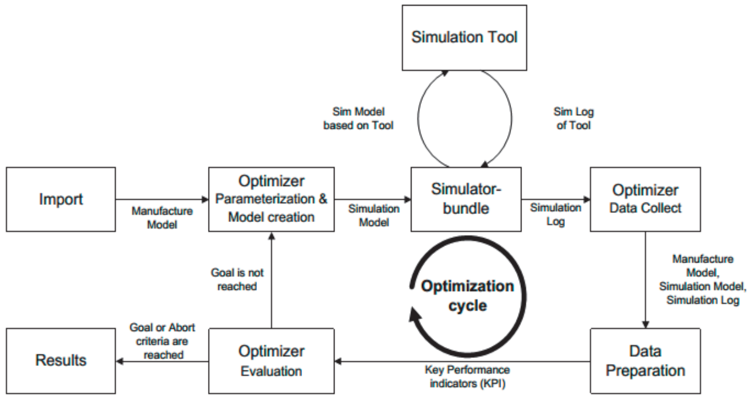


***Figure 2:*** *Simulation-based Optimization Cycle*

The structure of the optimizer module itself is divided into three almost independent sub modules. This segmentation allows easier maintainability, exchangeability and upgradeability.

First part of the optimizer is the parameterization, which includes the actual optimization algorithms. In the parameterization the attributes for the upcoming scenario are determined. These attributes are for example activity modes, earliest start and latest end dates, dispatching rules and schedule plans. For the first optimization cycle of an optimization run, default values are being used for these parameters. The parameterization of later scenarios is a result of the used optimization algorithm and is based on the information gained in previous runs.

The second part of the optimizer module is the model creation. Based on the factory model and on the attributes from the parameterization, a new scenario is created. This scenario is then stored in the database, where the simulation module can load it.

The evaluation is the third part of the optimizer module. After processing the simulation raw data in the data preparation module, an evaluation of whether another optimization cycle is necessary or not is made. Depending on the optimization target, several Key Performance Indicators are checked, (for instance SEMI (2004)). Based on these criteria it is decided whether the simulation result and therefore the latest scenario is acceptable or not. In addition the abortion criteria, defined as maximum number of optimization cycles or elapsed time, is checked. If a target or abortion criteria is met, the optimization is terminated. In the instance that neither target criteria nor abortion criteria are met, a new optimization cycle starts.

## 3 A Simulation Tool for Complex Assembly Lines

Over the last years using Simulation-based Optimization we worked with several general purpose simulation tools and we created for almost all of them suitable simulation modules for the second step of the optimization cycle in Figure 1. During the project, we realized that many of them are not efficient. The results are slow runtime, high development time/costs and sometimes unrealizable modules. For these reasons we created a specific custom-built simulator designed for this problem domain which uses the same internal meta-model as the platform. The tool supports a variety of real-world extensions and dedicated behaviour. This simulation tool is a further development of the simulation tool, which is presented in (Angelidis 2011). The main characteristics of the new simulation tool are:

- *Restart from Specific Time Point:* The goal of the optimization runtime is to gain in performance or simulation speed. As a part of the task, a mechanism was developed that allows performing execution not only from the beginning, but also from specific time points. Therefore, when a new simulation model is created according to the previous model, the tool can begin from the point of time where the first change appears, rather than starting from the beginning of the model. This method should reduce simulation runtime.
- *Activity Modes based on skills:* Activities do not have a fixed processing time but have various modes that describe their execution characteristics. The processing time is dependent on the amount of the skills of the resources. Activities have a high resource variation according to their execution and the resources are linked to a resource-qualifications table. More realistic models can be simulated. For the resource selection many rules are implemented, like choosing low cost resources, resources with a high number of availability, resources that probably will not be needed in the nearest future or resources with low utilization.
- *Internal and subcontracted workers:* The simulation model has resources defined as internal workers, who have a group of subcontracted workers. These can be used when the internal workers are scheduled but currently unavailable. This allows the optimizer to evaluate some manufacturing cost factors and analyse the due date violation over the resource utilization.
- *Resource Allocation Forecast:* Before the simulation core tries to distribute resources to the ready to start activities, it has to analyse the remaining processing time of all activities, which are already processing. If these remaining times are less than a given amount of time and their successors have a higher priority than the ready to start activities, then suitable resources will be reserved for these successors until the next event. It is a powerful way to help the activity schedule, while most of the general-purpose simulation tools only analyse the current time events ignoring the nearest future.
- *Workforce Resource Behaviour:* Two ways to execute an activity are implemented. When the state of allocated resource state changes, then the activity can find a new available or wait until the specific resources appears again.
- *Resource-Qualification Table with Dynamic State:* There is a Resource-Qualification table that allows the simulation core to select resources for every mode. This table is not static and can be changed according to many parameters that the optimizer defines (products near a due date, high utilization, high costs, etc.). It is a powerful extension to analyse the workforce behaviour under special circumstances.
- *Priority Rules:* We implemented more than forty well-known priority rules based on (Haupt (1989), Pinedo (2005), Pinedo (2008)). The calculation of the rules has a strong influence on the runtime of the simulation. Some activities have static values for some rules and we calculate them once (First In First out, Shortest/Longest Processing Time, Earliest Due Date etc.). On the other hand some rules are depended on network analysis or performance indicators and the values have to be calculated every time again as Critical Ratio, Least/Most Total

Successors etc. For that reason we implemented a dispatching rule framework for creating new rules with emphasis on optimized calculation.

## 4 Cost Function

As described before, the SBOP platform offers different algorithms for the scheduling of complex assembly lines. They are related to time and resource objectives, like mainly solution approaches in the field of resource constrained project scheduling problems. Since it is also necessary to complete projects cost efficient within due date we decided to develop a cost model, which serves as a new optimization goal. Some researchers present first solution approaches for cost oriented objectives. The cost analysis is mainly limited to activity costs and results are evaluated by self-generated, theoretical examples with only small and medium size instances. *Table 1* gives an overview of considered cost factors in resource scheduling problems.

***Table 1:*** *Published papers about cost factors (* only Penalty)*

| **Authors** | **Resource Cost** | **Overhead Cost** | **Bonus and Penalty Payments** | **Opportunity Cost (Unused Resources)** | **Sub-contracting Cost for External Resources** | **Other Cost Factors** |
|---|---|---|---|---|---|---|
| Resource Constrained Project Scheduling Problems and Generalization | | | | | | |
| Chen (1994) | X | — | X* | — | — | X |
| Ahn and Erenguc (1998) | X | — | X* | — | — | — |
| Salewski (1999) | X | — | — | — | — | — |
| Rummel et al. (2005) | — | — | X* | — | — | X |
| Ke and Liu (2005) | X | — | — | — | — | — |
| Varma et al. (2007) | X | — | — | — | — | — |
| Voss and Witt (2007) | — | — | — | X | — | X |
| Tseng (2008) | — | — | X | — | — | — |
| Liu and Zheng (2008) | X | — | — | — | — | X |
| Multi–Mode Resource Constrained Project Scheduling Problems | | | | | | |
| Santos and Tereso (2011) | X | — | X | — | — | — |

*Figure 3* shows the new developed cost function TC(x). The approach considers all cost factors which are influenceable by scheduling, like resource cost, opportunity cost (unused resources), subcontracting cost, bonus and penalty payments and

assembly overhead cost. This allows us a deep cost evaluation as well as an economic comparison of different production scenarios.

$$
\begin{aligned}
&\textit{Minimize}\ \mathrm{TC}\ (\mathrm{x}) \\
&\text{Subject to:} \\
&TC(x) = \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{k=1}^{K}\sum_{m=1}^{M_{ij}} x_{ijm}\left[\left(c_k^r * D_{ijm}^r + c_k^o * D_{ijm}^o + c_k^s * D_{ijm}^s\right) * r_{ijmk}\right] \\
&+ \sum_{t=1}^{T}\sum_{k=1}^{K} c_o^k * Max\left[0,\left(R_{kt} - \sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{m=1}^{M_{ij}} x_{ijmt} * r_{ijmk}^t\right)\right] \\
&+ \sum_{t=1}^{T}\sum_{k=1}^{K} c_e^k * \mathrm{Max}\left[0,\left(\sum_{i=1}^{I}\sum_{j=1}^{J_i}\sum_{m=1}^{M_{ij}} x_{ijmt} * r_{ijmk}^t - R_{kt}\right)\right] \\
&+ \left|\bigcup_{i=1}^{I}\bigcup_{j=1}^{J_i} t_{ij}^a\right| * c_a + Max(U_i) * H \\
&+ \sum_{i=1}^{I} Max\left[0, v_i\frac{U_i - g_i}{n}\right] - \sum_{i=1}^{I} Max\left[0, b_i\frac{g_i - U_i}{n}\right]
\end{aligned}
$$

***Figure 3:*** *Manufacture cost function*

**Let:**

| | |
|---|---|
| $i$ | order index i = 1, 2, ..., I (I ... total number of orders) |
| $j$ | activity index j = 1, 2, ..., Ji (Ji ... total number of activities and the index of the last activity j of order i) |
| $k$ | resource type index of the renewable resource k = 1, 2, ...,K |
| $m$ | mode index m = 1, 2, ..., Mij (Mij ... total number of modes and index of the last mode of activity j of order i) |
| $t$ | serial time index t = 1, 2, ..., T (T ... end of planning horizon – actual due date of the last order: T = max[Ui]) |
| $TU$ | time-unit, to count the cost factors |
| $b_i$ | bonus for early completion – marginal gain from early completion |
| $fst_i$ | earliest start date |
| $g_i$ | specified delivery date |
| $H$ | overhead costs |
| $M_{ij}$ | amount of feasible modes m of an order i for activity j |
| $n$ | number of time units t to count bonus and penalty |
| $U_i$ | actual due-date of order i |

| | |
|---|---|
| $v_i$ | penalty for exceeding the specified delivery date – marginal loss from late completion |
| $c_o^k$ | opportunity costs of an unused resource unit of type k per time-unit t |
| $c_a$ | fixed general and administrative costs per time unit t to run the assembly line |
| $c_e^k$ | lease cost for an external resource unit of type k per time unit t |
| $c_k^o$ | resource cost for the usage of one resource of type k during overtime |
| $c_k^r$ | resource cost for the usage of one resource of type k during regular working hours |
| $c_k^s$ | resource cost for the usage of one resource of type k during special shifts |
| $f_{ij}$ | simulated earliest start date of activity j of order i |
| $l_{ij}$ | simulated latest start date of activity j of order i |
| $R_{kt}$ | available resource capacity of type k within time unit t |
| $r_{ijmk}^t$ | required amount of resources of type k for activity j of order i |
| $D_{ijm}^o$ | processing time of the activity j of order i in mode m during overtime (considering all resources and depending on the determined mode) |
| $D_{ijm}^r$ | processing time of activity j of order i in mode m during regular working hours (considering all resources and depending on the determined mode) |
| $D_{ijm}^s$ | processing time of the activity j of order i in mode m during special shifts (considering all resources and depending on the determined mode) |
| $t_{ij}^a$ | time unit t, in which an activity j of order m in mode m is actually processed |
| $x_{ijmt}$ | a binary variable, of value 1 if activity j of order i in mode m is finished in time unit t, and 0 otherwise |

## 5 An Optimization Heuristic

Analysing the work of Majohr (2008) we found several points to improve this heuristic strategy in a way that they yield better optimization results with less optimization cycles required. We present a three-phase optimization strategy:

1. Feasibility tests of the production plan with minimal processing times and maximal personnel resources.
2. Iterative reduction until reaching minimal personnel capacities.
3. Maximal cover of the present personnel capacity curves.

In the first phase we choose the fastest modes for activities, the earliest start dates and latest end dates for orders, products and activities, as well as artificial schedule plans for resources. These schedule plans can exist or contain endless, maximum amount of resource units available in the resource pool. Under the assumption that the fastest modes will lead to the earliest possible completion time, we use this phase to validate, whether it is possible to create scenarios without due date violation under given resource pool size and timeframe. If the scenario fails at this stage, it will probably not be possible to create a valid scenario. In that case we can use our metamorphosis algorithm, which is described later. In phase two we start with our actual optimization. The goal is to reduce the utilization of the resources without violating any due date of the orders, products and activities. The challenge is to complete production with as little slack as possible, while not violating the due dates. In our modelling approach we cannot influence the operation time of our orders directly, but we can influence the operation time of the activities by changing their modes. The method in (Majohr 2008) tries to distribute the slack of the order to its products and to its activities. In combination with the raw process time (RPT) of the activity in the previous optimization cycle, this share of slack is used to determine a new target raw process time. Based on this, we look for modes which meet this target RPT. These modes have longer RPT and lesser resource consumption. Using these modes, the activities will be slowed down, as will be the orders and subsequently their slack will be reduced. The most important part of this method is the way the slack is distributed among the activities. In (Majohr 2008) they simply spread the slack on the activities according to their share on the order's operation time. Longer activities get more share of the slack and shorter activities get less share of the slack. We extended this algorithm (Majohr 2008) by focusing especially on activities on the critical path. These activities are directly responsible for the completion date of the order. To determine whether an activity is critical or not, the Extended Critical Path Method of (Angelidis et al. 2012b) is used. At first, slack is distributed on these activities by their share on the order's operation time. Altering the operation time of critical activities alone, would be enough to influence the completion time of the order. In a second step, we also have to slow down non-critical activities as they allocate important resources as well. In contrast to the critical ones, we can slow down these activities by a much larger scale, because they are not directly tied to the order's completion date. In addition the separation of critical and non-critical activities is also relevant in case of a negative slack, which means a due date violation. When we want to speed up the order, we have to know exactly which activities are responsible for the completion time of the order and accelerate just these activities, to prevent unnecessary resource consumption. Another improvement we made is bound to the determination of the new raw process time of activities. In (Majohr 2008) they accounted the distributed slack directly as additional time for the raw process time. Unfortunately this does not consider times without production, like free shifts or weekends. In our approach we compare the activities' raw process times to their operation times and create a ratio on this. Every slack, which is distributed to an activity, will be reduced by this factor to create a more realistic target raw process time. In some special cases where the activities and the product have individual due date, the calculation of a minimum slack without any due date violation is almost impossible. On such cases we use a promising solution based on self-organization approach (Angelidis et al. 2012a). Whenever one of the other algorithms is not able to create a valid scenario, the metamorphosis algorithm enables to commit changes to the model, which cannot be done by the traditional algorithms, such as changing shift plans or adding resources. This is inspired by real production lines, where overtimes, additional shifts and external resources are used to compensate an over amount of work. To solve such a problem, we use several escalation steps (extend the duration of normal shifts or add new shifts to night or weekend, add additional resource units to the shifts, up to the maximum of units in the resource's pool, add additional resource units to the resource pool). Obviously in most cases it would not make sense to add expensive resources like operating facilities, but it is a usual practice to add personnel based on the cost function. In the third part which is time consuming, we try to maximise the cover of the present personnel capacity curves based on the best scenarios of phase two. We analyse the time periods of the unused resources and try to minimize them according the used activities on that time. If the cover maximisation is not possible we try to keep almost constant the amount of the used resources. Because this is steps needs a lot of cycle, we use our special simulation tool to start the simulation of the scenarios from specific time points.

## 6 Results and Conclusion

We run several tests on the algorithms to quantify the quality of the optimization methods. The slack minimization algorithm was able to reduce the slack by 90%, while the method introduced in (Majohr 2008) reduced the slack by 88% of a comparable scenario. However our approach needed just 2 optimization cycles, while the method of (Majohr 2008) needed 6 cycles. This is mainly due to the improved slack distribution using the critical path and the removal of the effects of weekends, free shifts and breaks on the RPT calculation. Figure 4 shows the utilization values of 5 processing facilities over 5 optimization cycles and a target utilization of 60%, for the utilization balancing algorithm on the left and the combined algorithm on the right. Please note the straight development of the utilization values towards the target using the balancing approach. In comparison the utilization values in the combination method show a swinging behaviour, while the algorithm tries to reach the target value and to minimize the slack at the same time. In the given example the slack is reduced by 88%, while the method of (Majohr 2008) just reduced the slack by 78% in a comparable scenario. Currently our research focuses on the third phase.

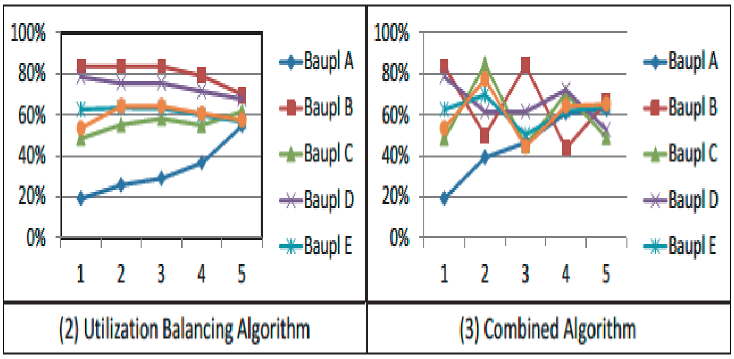


*Figure 4: Optimization with Workcentre Utilization Goal 60%*

## Promotion Information

The presented cost function is a result of a current research project (393 ZBR): “Simulationsbasierte Prozesskostenrechnung zur Bestimmung kostenminimaler

Ablaufalternativen in der Montageplanung bei KMU" It is funded by the Bundesministerium für Wirtschaft und Technologie (Federal Ministry of Economics and Technology) of the Arbeitsgemeinschaft industrieller Forschungsvereinigungen "Otto von Guericke" - AiF (Federation of Industrial Research Associations "Otto von Guericke" - AiF) on behalf of the Bundesvereinigung Logistik e.V. - BVL (German Logistics Association, incorporated association - BVL) and is being processed over the period 11/2011 to 09/2014.

## References

Angelidis E.; Pappert F. M.; Rose O.: A Prototype Simulation Tool for a Framework For Simulation-based Optimization of Assembly Lines. In: S. Jain, R.R. Creasey, J. Himmelspach, K.P. White und and M. Fu (Hg.): Proceedings of the 2011 Winter Simulation Conference. Institute of Electrical and Electronics Engineers, Inc. Piscataway, New Jersey, S. 2383–2394.

Angelidis E.; Bohn D.; Rose O. (2012a): A Simulation-based Optimization Heuristic Using Self-Organization for Complex Assembly Lines. In: C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose und A. M. Uhrmacher (Hg.): Proceedings of the 2012 Winter Simulation Conference. Institute of Electrical and Electronics Engineers, Inc. Piscataway, New Jersey.

Angelidis E.; Naumann A.; Rose O. (2012b): An Extended Critical Path Method for Complex Assembly Lines. In: G. Lim und J.W. Herrmann (Hg.): Proceedings of the 2012 Industrial Engineering Research Conference.

Ahn T.; Selcuk E. A.: The resource constrained project scheduling problem with multiple crashable modes: A heuristic procedure. In: European Journal of Operational Research 107 (2), (1998) S. 250–259.

Chen V. Y.X.: A 0-1 goal programming model for scheduling multiple maintenance projects at a copper mine. In: European Journal of Operational Research 76 (1), (1994), S. 176–191.

Evers K.: Simulationsgestützte Belegungsplanung in der Multiressourcen-Montage. Dissertation, University Hannover, 2002.

Haupt, R.: A survey of priority rule-based scheduling. Operations-Research-Spektrum 11 (1): 3–16. 1989

Ke H.; Liu B.: Project scheduling problem with stochastic activity duration times. In: Applied Mathematics and Computation 168 (1), (2005), S. 342–353.

Liu Z.; Zheng Y.: Resource-constrained multiple projects scheduling with the objective of minimizing activities cost. In: Control and Decision Conference, 2008. CCDC 2008. Chinese, S. 1027–1032.

Majohr.M. F.: Heuristik zur personalorientierten Steuerung vom komplexen Montagesysteme. PhD thesis, Dresden University of Technology, 2008

Pappert F. M.; Angelidis E.; Rose O.: Framework for simulation based scheduling of assembly lines. In: Proceedings of the 2010 Winter Simulation Conference. Institute of Electrical and Electronics Engineers, Inc. Piscataway, New Jersey, S. 1690–1968.

Pinedo, M. L.: Planning and Scheduling in Manufacturing and Services (Springer Series in Operations, Research and Financial Engineering). Springer. 2005

Pinedo, M. L.: Scheduling: Theory, Algorithms, and Systems. 3rd ed. Springer. 2008

Rummel J. L.; Walter Z.; Dewan R.; Seidmann A.: Activity consolidation to improve responsiveness. In: European Journal of Operational Research 161 (3), (2005), S. 683–703.

Salewski F.: Heuristiken zur Dienstplanung bei flexibler Personalkapazität. In: OR-Spektrum 21 (3), S. 361–379.

Santos M. A.; Tereso A. P.: On the Multi-Mode, Multi-Skill Resource Constrained Project Scheduling Problem - Computational Results. In: 1st International Conference on Project Economic Evaluation, ICOPEV'2011.

SEMI: "EMI E10-0304E: Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)", 2004.

Tseng C. C.: Two Heuristic Algorithms for a Multi-Mode Resource-Constrained Multi-Project Scheduling Problem. In: Journal of Science and Engineering Technology 4 (2), 2008, S. 63–74.

Voß S.; Witt A.: Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application. In: International Journal of Production Economics 105 (2), (2007), S. 445–458.

Varma V. A.; Pekny J. F.; Blau G. E.; Reklaitis G. V.: A framework for addressing stochastic and combinatorial aspects of scheduling and resource allocation in pharmaceutical R&D pipelines. In: Computers & Chemical Engineering 32 (4 - 5), (2208), S. 1000–1015.