

## A SIMULATION BASED OPTIMIZATION ALGORITHM FOR SLACK REDUCTION AND WORKFORCE SCHEDULING

Daniel Noack  
Oliver Rose

Institute of Applied Computer Science  
Dresden University of Technology  
Dresden, 01062, GERMANY

### ABSTRACT

In an assembly line with high labor proportion, the workforce planning and scheduling is a very complex problem. At the background of increasing labor costs, it is very important to increase workforce efficiency. It is essential for companies to remain competitive on global markets. Increasing efficiency is our motivation to work on simulation-based workforce scheduling for complex assembly lines. In this paper, we will focus on the heuristic algorithm in our simulation-based optimization approach. The objective is workforce quantity and slack reduction. To improve the objective, an algorithm assigns the number of workers for activities, scheduled in the simulation run. We will present three different strategies implemented in these optimization algorithms. They basically use the performance indicator slack time, work center utilization and a mix of both parameters. We will compare the algorithms according to their achieved objective and the required computation time.

### 1 INTRODUCTION

To make a value add operation in an assembly line, you need to bring together a number of workers, the work center and an assembly part at the same time. If one element is missing, the whole process will not continue and no value will be added. The overall complexity of this scheduling problem is very high, because of the high interaction of all elements in an assembly line. There are numerous work centers with parallel machines, groups of workers with different skills and a lot of different parts to assemble. In an assembly line, each part has a well defined sequence of production steps, even with merging and splitting activities. To run on a certain work center, each activity requires workers with different skills. So, if one small element is being changed in the schedule, like the assignment of one worker to another operation, it affects all related elements. One activity will be delayed, another activity will be accel-

erated. All following activities on these machines will be delayed or accelerated as well. At the end this small decision affects the whole assembly line and its performance.

The workforce scheduling problem contains two major components, the Job Shop Scheduling Problem and the Personnel Allocation Problem.

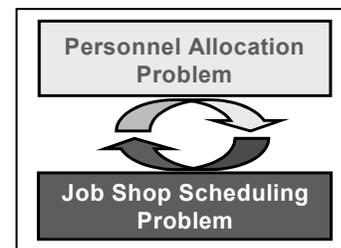


Figure 1: Problem Description

The Job Shop Scheduling Problem defines the time for activities given on certain work centers. It also affects the Personnel Allocation Problem, because it defines the time when the workers are required at particular work centers. The Personnel Allocation Problem assigns a number of workers for the activities. It also effects the Job Shop Scheduling Problem. If the number of workers assigned to an activity gets changed, the processing time also changes. For a detailed description of the problems, see (Pinedo 1999).

Traditionally the whole schedule was created by hand. The foreman needs lots of experience to assign workers. Because of the high complexity in a large assembly line, a manually created schedule does not always deliver good solutions. In daily operations, a manual planning process will also consume a lot of time. So there is a huge potential of automating the planning process and achieve improvements in factory performance. The main target of a software solution is to increase workforce efficiency and decrease workforce overcapacities. It will also improve constraint satisfaction and provide planning security. To

achieve user acceptance a low computation time is required. The whole problem solution has to be finished within a few minutes.

Our simulation-based optimization approach is capable to provide such a solution. The focus of this paper is the heuristic optimization algorithm. It is a major part in the simulation-based optimization approach.

## 2 PROBLEM DESCRIPTION

In the following section, a detailed scenario will be explained for a better understanding of the problem in aviation industry. The machine pool in this real world scenario contains about 30 work centers with parallel machines and 5 transport resources. Only a limited number of the buffer spaces are available. In total more than 200 workers with different skills are organized in 15 different groups. Each worker has to be assigned to a work center, see Figure 2. The whole model requires detailed resource data to model multiple shifts, worker break times and scheduled downs for work centers. A typical planning horizon in an assembly line in aviation manufacturing is more than two month. During this time more than 15 orders are release to the line. An order contains a set of activities, to create a product for a customer. Each order has an individual release date and a due date. Orders can have different routes in the assembly process. The route consists of a well defined sequence of more than 200 activities. An activity contains the job definition. It defines how long one part is assembled on a work center with a given number workers with defined skills. Also complex splitting and merging activities with multiple successor and predecessor activities are possible. Some activities take parallel or alternative routes. All of these activities have to be scheduled. In the assembly line various alternative activities exist where the only difference is the number of workers and the activity duration. These activities are called activity modes. All these activities have to be selected and scheduled as well.

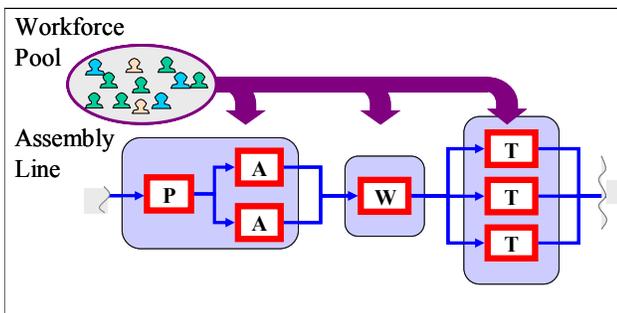


Figure 2: Assembly line

In an assembly line there are a lot of additional constraints. There are maximum time bounds for two or tree activities in a row. Some activities have to be processed on

the same resources, even if alternative resources are available. Another constraint is the limited buffer space. The reason is that the physical parts are so huge that limited buffer space and blocking of work centers becomes very important. A work center is blocked if a part cannot be removed, because the following work center is not available and there is not enough buffer space between both work centers. We need a high detail model to consider blocking, buffers and transportation.

The overall objective for this scenario is to improve the factory performance. The factory performance depends on various performance indicators like work in progress, cycle time, utilization and so on (Hopp and Spearman 2000). Objectives are to reduce overcapacity by increasing efficiency. That means reducing the idle time of the workforce and the work centers. Usually in a complex assembly line there is much overcapacity of worker resources. The reason is to guarantee all due dates. With a better schedule and higher planning security we intend to achieve all objectives like keeping the due dates with fewer resources.

## 3 SOLUTION APPROACH

To find a reasonable solution, a simulation-based optimization approach is used, see (Majohr 2008). The simulation model represents a real world facility. The optimization algorithm changes settings and parameter of the simulation model, see Figure 3. The control vector  $x$  defines which changes will be applied during the simulation. The simulation generates a deterministic schedule. Based on this schedule we compute an objective value  $C(x)$ . Depending on this objective, the optimization searches for new simulation settings to generate a better objective value. This is an iterative process with small step by step improvements.

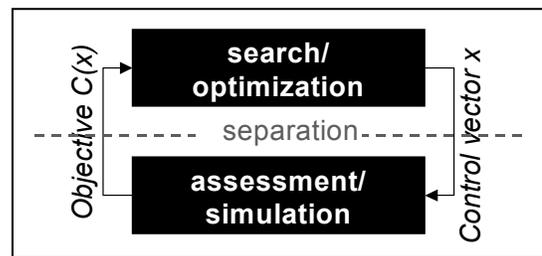


Figure 3: Simulation-based Optimization

Actually various simulation tools with optimization packages are available, see (Law and McComas 2002). These approaches were applied successfully, for example in a semiconductor manufacturing (Horn et al. 2006) or in project management (Hanna and Ruwanpura 2007). For this solution approach a heuristic search algorithm is used for the optimization. Search methods like linear programming (Haas et al. 2002) or genetic algorithm (Hegazy 1999) were applied successful, too. But for this problem

they are not applicable because of the huge problem size and the computation time bounds. Pilot tests indicate that it would require too much time, to generate a reasonable solution.

To apply a simulation-based optimization approach to an assembly line we need both, a simulation model and an optimization algorithm.

The simulation model represents constraints, dispatching, buffer allocation and so on. In every run the Simulation generates a deterministic schedule. For this schedule, the objective  $C(x)$  will be computed. In this particular scenario the objective is to minimize the slack, keep all due dates and minimize workforce overcapacity. The simulation solves the Job Shop Scheduling Problem with dispatching rules.

The optimization algorithm will not change the sequence of the jobs directly. The optimization algorithm only solves the workforce allocation problem by changing the number of workers assigned to an activity. For all activities in the Control vector  $x$ , the optimization selects one activity mode from a set of available activity modes. In Figure 4 we depict different available activity modes for one activity. Currently mode 4 is selected. All activity modes differ only in the execution time for an activity and in the required number of workers. For the numbers of workers only integer values are used.

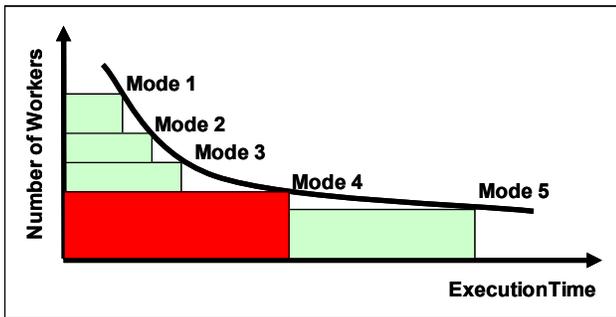


Figure 4: Activity Modes

The optimization algorithm chooses activity modes with a short execution time but it will require a large number of workers. On the other hand it can choose activity modes with only a few workers, but it will take quite a long time to finish. In Figure 5 we compare three different options to execute an order. In option A the maximum number of workers are assigned to an order. Each activity will be executed very fast. The slack becomes very large. In option B the number of workers is reduced for activity two. The effect is that activity three starts later and the slack of the order becomes smaller. In option C the number of workers is reduced again for activity two and three. The whole order is delayed and cannot meet its due date.

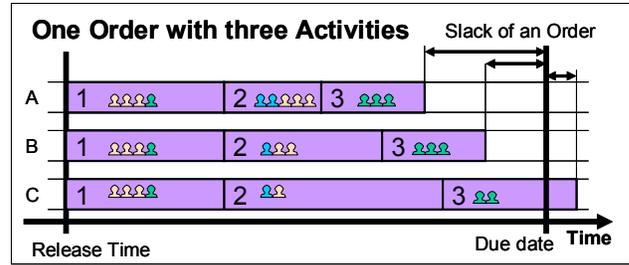


Figure 5: Activity Modes

To obtain a better understanding about the complexity of the optimization, we provide a rough estimation about the problem size. The whole model contains more than 500 activities with different activity modes. This leads to 500 decision variables in the optimization. On average the optimization can choose between six activity modes for one activity. These are 6 to the power of 500 different combinations. A full enumeration of this problem on the best high performance computers will take several years of computation time. The practical requirement for our solution algorithm is to compute a valid solution in only a few minutes. Our target is to find a minimum number of workers in less than two minutes.

#### 4 OPTIMIZATION STRATEGIES

As mentioned above the heuristic algorithm is embedded in a simulation-based optimization procedure. The whole optimization process has other elements like a termination criteria, an objective function and modules to control the optimization process. But the focus of this paper is the optimization strategy itself. We describe the way how to change simulation model settings between two iterations.

Basically the optimization starts with choosing the maximum number of workers for all activities. The first target is to generate a valid schedule. If it's not possible to generate a valid schedule with the maximum number of workers, the optimization stops immediately. The reason is wrong user input data, like false due dates for an order. In most cases a valid schedule is available after the first iteration. Then the optimization algorithm minimizes the amount of workers step by step. It chooses activity modes with longer execution time and less workers. All changes in the simulation settings are based on simulation settings from the previous iteration. In the following section different strategies for workforce reduction are presented.

##### 4.1 Minimizing the Number of Resources by Slack Reduction

The main strategy in this algorithm is to increase the processing time by using the slack time. The algorithm expands the whole order by selecting activities with a lower number

of workers. In Figure 6 we provide an example for this approach.

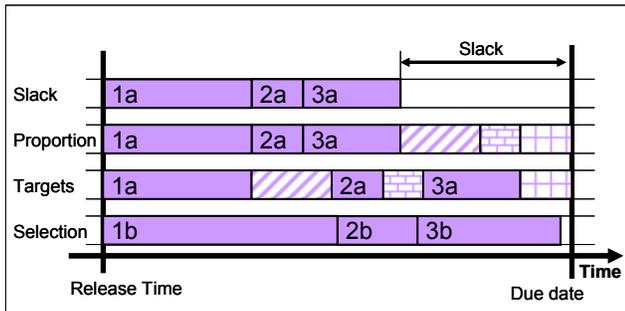


Figure 6: Slack Reduction

The first step computes the slack of an order. The slack is the duration from the completion time to the due date. The second step calculates the proportion of the slack time for each activity. Basically the algorithm computes the proportion of each activity on the makespan of an order. This proportion is multiplied by the slack of the order. The result is the time you want to expand the duration of each activity. In the third step, this result will be added to the activity duration. The algorithm generates the target duration of each activity. The fourth step selects the activity mode matching the target duration best. As mentioned before, only discrete activities with integer number of workers are available. So, the algorithm selects the activity mode with the minimum difference between target duration and the duration of this activity mode.

#### 4.2 Minimizing the Number of Resources by Utilization Balancing

In this algorithm the strategy is to balance the utilization of all work centers. The main idea is to increase the number of workers on bottleneck work centers and reduce them on work centers with large idle times. In Figure 7 we depict an example of the work center utilization for work center A, B, C and D. Work center A is the bottleneck, having the highest utilization. By assigning two more workers to work center A the utilization is being reduced. For work center C and D the utilization increases because the number of workers is reduced by one on each work center. The reason is that the more workers are assigned to a work center the shorter is the execution time of an activity and the more activities can run on this work center. In this example the work center utilization is almost the same after the optimization.

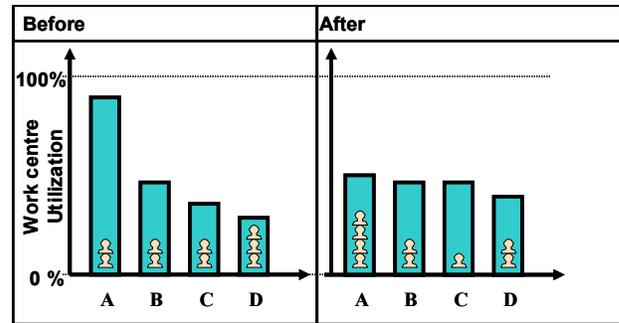


Figure 7: Work center utilization balancing

The algorithm sorts all work center with respect to their utilization. It reduces the number of workers on activities running to work centers with low utilization. If the slack becomes negative, more workers will be assigned on work center with high utilization. The algorithm can be parameterized by a step width. Here it is the number of work centers, with changing activity modes. The step width of activity modes can change too. In a list of activity modes, sorted by the number of required workers, the algorithm can choose the neighbor of the current activity mode. But it is also possible to choose activity modes that are 2 or 3 steps away from the currently selected activity mode.

#### 4.3 Minimizing the Number Resources by Slack Reduction and Utilization Balancing

The third strategy combines the first and second algorithm. It considers both, the slack time of an order and the work center utilization. The objective of this strategy is to minimize slack time and balance work center utilization at the same time. In Figure 8 we show the workforce assignment scheme. If the work center utilization is very low and the slack of an order is very large, the workforce reduction is very high. In the best case, the utilization is high and the slack is positive and small. Then no changes will be applied. If the slack of an order is negative the number of workers has to increase. The higher the utilization of a work center, the more workers it will obtain.

		Work centre Utilization	
		Low Utilization	High Utilization
Slack of an Order	Large Positive Slack	High Workforce Reduction	Low Workforce Reduction
	Small Positive Slack	Low Workforce Reduction	No Workforce Changes
	Large Negative Slack	Low Workforce Increase	High Workforce Increase

Figure 8: Workforce assignment scheme

This algorithm starts with the same computation steps as both previous algorithms. It sorts the work centers by their utilization and computes the slacks of all orders. Depending on its parameterization the algorithm changes activity modes on a predefined number of work centers for a predefined percentage of the slack time.

### 5 EXPERIMENTAL RESULTS

This section contains the experimental results for a real world scenario. In the first part the relation between slack and workforce quantity is presented. In the second part the comparison of all three algorithms is shown.

Figure 9 depicts the relation between slack time and workforce quantity. The optimization algorithm reduces the number of workers and the slack between two iterations. Both curves in Figure 9 are related. By reducing the number of workers, the slack time is also reduced.

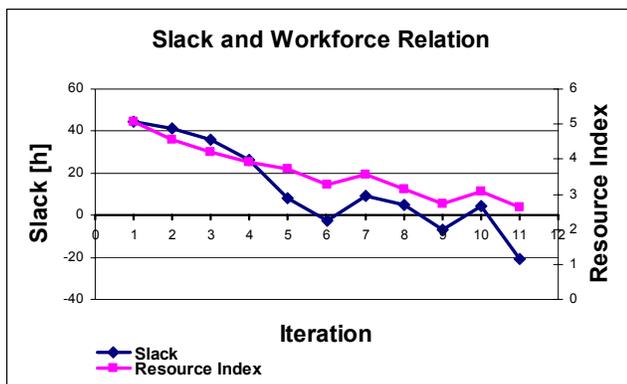


Figure 9: Slack and Workforce Relation

The resource index is used instead of the average number of required workers. It is a better performance indicator because the average number of required workers does not change so much for the same time horizon. If the

number of workers is reduced for one activity, then the execution time of this activity will become longer. So, for the same time horizon the average number of workers is pretty much the same.

In Figure 10 we compare the slack reduction of all three algorithms. As mentioned before, the first iteration is running with the maximum number of resources. For the first iteration the slack has a maximum of 44 hours. All three algorithm starts at the same position. In each iteration the algorithms start to reduce the number of workers and minimize the slack. It's interesting to compare all three algorithms.

The first algorithm is minimizing the slack only. It reaches its objective very fast. After tree iterations it the minimum slack is less than two hours. It remains very stable during further iterations.

The second algorithm only considers work center utilization. The slack converges very slowly. After ten iterations it is close to the optimum slack of zero. That is too long according to the computation time limit. In the simulation-based optimization approach one iteration takes about 20 seconds. To beat the target of two minutes, only six iterations are possible. The second algorithm does not get very close to the target at the sixth iteration.

The third algorithm is considering slack and work center utilization. The slack reduction is a little bit slower than in the first algorithm but still faster than in the second one. This algorithm can still meet the target of six iterations to find a reasonable schedule with a small slack.

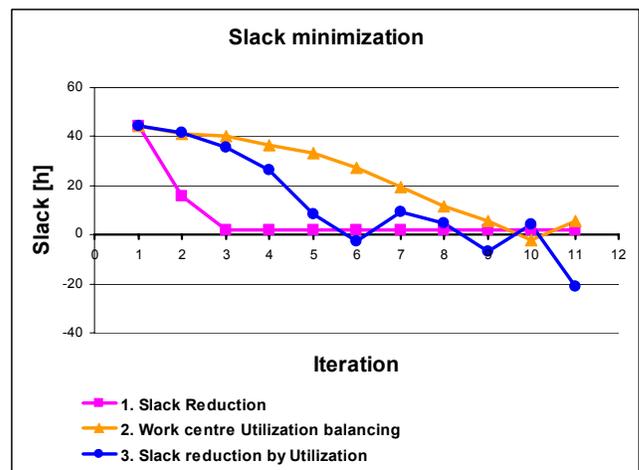


Figure 10: Slack minimization

We find that in some iterations the schedule has a negative slack. The algorithm assigns more workers in the following iteration to obtains a positive slack.

In this particular scenario the expectations are to obtain a better performance by considering the utilization of work centers. But the difference in work center utilization between algorithm one and three is only marginal. The dif-

ference in the bottleneck utilization is only two percent for this particular scenario.

## 6 CONCLUSIONS

We present three heuristic algorithms for a simulation-based scheduling system. We show that the implemented algorithms are capable to improve the performance of a complex assembly line. We compared these algorithms in a real world scenario to show the advantages and disadvantages. All algorithms are able to reduce the slack and workforce quantity. They differ in the final objective and in computation time to achieve this objective. Two out of three algorithms are capable to reach their objective within the time limit.

## 7 OUTLOOK

In the future there are two major goals – to increase the level of details in simulation model and to realize the resource balancing in the optimization.

Adding details to the simulation means to add a more detailed shift and break time model. It is also required to add better decision rules to the simulation like dispatching rules, buffer and tool allocation rules.

The resource balancing will be the most important part in the optimization and it will be the most interesting part from an academic point of view. Currently the heuristic algorithm just reduces the workforce quantity. But after this step, it is important to balance the number of required workers with the number of available workers. Then it will be possible to compare real world performance with the results from the generated schedule.

## REFERENCES

- Haas C. T., D. P. Morton, R. L. Tucker, J. E. GOMAR, R. K. Terrien. 2002. Assignment and Allocation Optimization of a Partially Multiskilled Workforce. *Journal of construction engineering and management* 128 (2): 103-109.
- Hanna M., J. Y. Ruwanpura. 2007. Simulation Tool for Manpower forecast Loading and Resource leveling. In *Proceedings of the 2007 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.
- Hegazy T. 1999. Optimization of Resource Allocation and Levelling using Genetic Algorithms. *Journal of construction engineering and management* 125(3):167-175.
- Hopp W. J., M. L. Spearman. 2000. *Factory physics*. 2nd ed. New York: Irwin/McGraw-Hill.
- Horn S.,G. Weigert, S. Werner, T. Jähnig. 2006. Simulation based scheduling system in a semiconductor backend facility. In *Proceedings of the 2006 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.
- Law A. M., M. G. McComas. 2002. Simulation based Optimization. In *Proceedings of the 2002 Winter Simulation Conference*. Piscataway, NJ: Institute of Electrical and Electronics Engineers.
- Majohr M. F., O. Rose. 2008. A Simulation-based workforce scheduling heuristic for complex assembly lines. In *Proceedings of the 2008 Operational Research Society Simulation Workshop*.
- Pinedo M., Xiuli Chao. 1999. *Operations Scheduling with Applications in Manufacturing and Services*. Irwin McGraw-Hill, Boston, MA.

## AUTHORS BIOGRAPHIES

**DANIEL NOACK** is a PhD student at Dresden University of Technology. He is a member of the scientific staff of Prof. Dr. Oliver Rose at the Chair of Modeling and Simulation. He received his M.S. degree in computer science from Dresden University of Technology. His e-mail address is [<daniel.noack@inf.tu-dresden.de>](mailto:daniel.noack@inf.tu-dresden.de).

**OLIVER ROSE** holds the Chair for Modeling and Simulation at the Institute of Applied Computer Science of the Dresden University of Technology, Germany. He received an M.S. degree in applied mathematics and a Ph.D. degree in computer science from Würzburg University, Germany. His research focuses on the operational modeling, analysis and material flow control of complex manufacturing facilities, in particular, semiconductor factories. He is a member of IEEE, INFORMS Simulation Society, ASIM, and GI. Web address: [<www.simulation-dresden.com>](http://www.simulation-dresden.com).